Examining the Impact of an Information Retrieval Pattern Language on the Design of Information Retrieval Interfaces

A Thesis

Submitted to the Faculty

Of

Drexel University

by

Christine Elizabeth Wania

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy

May 2008

Acknowledgements

I realize that I would not be writing these words without the support and guidance of so many people. I would like to thank all those people, even those not listed here, who have contributed to this work in any way.

Michael Atwood, Ph.D., my dissertation advisor, thank you for your invaluable guidance, support, and encouragement. Thank you for introducing me to the literature on pattern languages and design theory.

I would like to thank the members of my committee, each inspiring me, challenging me, supporting me, and contributing in their own way, Denise Agosto, Ph.D., Thomas Hewett, Ph.D., Gregory Hislop, Ph.D., and Susan Wiedenbeck, Ph.D. Thank you for all of your valuable thoughts and feedback.

Thank you to the faculty and staff at the College of Information Science and Technology, especially David Fenske, Ph.D., for the financial support. Kate McCain, Ph.D., thank you for introducing me to the world of bibliometrics.

Thank you also to the Trustees of the Phoebe W. Haas Fellowship for the generous fellowship award in my last year of studies.

Anthony Scimè, Ph.D., thank you for inspiring me to pursue a doctoral degree.

I would like to thank you my family and friends for their endless encouragement and support throughout my years in this program. Thanks especially to my Mom and Dad, Joan and Donald Wania, for teaching me the importance of lifelong learning. Dan, thank you for always being patient and understanding throughout these past four years. Thank you also to Anne, Jamey, Devon, Emma, Michelle, Mike, Rob, Jamie, Olivia, Jennifer, Christopher, Kathleen, Theresa, Jeannine, Joe, Tyler and Becky for all your support throughout the years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Examining the Impact of an Information Retrieval Pattern Language
on the Design of Information Retrieval Interfaces
Christine E. Wania
Michael E. Atwood, Ph.D.

For more than two decades much of the pattern language literature, within the field of Human Computer Interaction (HCI), has focused on the possible benefits pattern languages may provide, but there has been very little empirical work to support these claims. It has been suggested that interaction patterns or pattern languages in HCI may address some of the problems inherent in designing interactive systems by supporting reuse, capturing design knowledge, enabling the sharing of design knowledge, and facilitating communication among designers and users. This study examined the impact of a pattern language on the design of information retrieval interfaces, in terms of the quality of the interfaces and the time to design the interfaces. Participants created paper and pencil interfaces based on the given design task. Participants were exposed to either a pattern language, guidelines, or no structuring technique. There were no statistically significant differences between the three groups in terms of the quality of the interfaces and time to design the interfaces.

The results of this study suggest that the value of pattern languages in HCI may not be in reuse, at the early stages of design, or in terms of the quality of the resulting designs, in domains familiar to designers. Although there was no apparent impact of the pattern language on the early stage designs, the results of a follow-up study suggest there is a significant correlation between the existence of patterns in commercial systems and the overall usability of those systems. Therefore, we suggest that we, as a community, very closely examine the current state of pattern languages in HCI before continuing to move forward. As a

community, we need to shift our focus away from discussing the possible benefits of pattern languages and trying to build pattern collections. And instead, focus on trying to fully understand the value of pattern languages in HCI. In doing so, the HCI community, will then begin to see the benefits from all the great efforts in this area.

1. INTRODUCTION

Design is a word we all understand in some way. The process of design, on the other hand, is one aspect of design, among many others, which we may not fully agree upon or understand. Design has been described by some as an art and by others as a science. Although we, as a design community, may not agree upon what design is, we do agree that there are many problems inherent in design. Some of the problems include defining the problem, deciding what to design, and communicating with people from different disciplines.

The design of interactive systems is difficult because many of the problems which interactive systems address are complex, ill-structured (Simon, 1984a), or wicked (Rittel & Webber, 1984). Therefore, it is necessary to incrementally grow, not build, software systems (Brooks, 1987). There are various Human Computer Interaction (HCI) design and evaluation techniques used by researchers and designers to aid in the evolutionary process of information systems design. A variety of techniques are discussed in HCI literature but, the relationships between many of these methods are not well understood. Some of the more common design methods and techniques used in HCI include participatory design (Kyng, 1991), user-centered design (Norman, 2002), interaction design (Preece, Rogers, & Sharp, 2002), scenario based design (Rosson & Carroll, 2002), contextual design (Beyer & Holtzblatt, 1998), and paper prototyping (Snyder, 2003). Some common evaluation methods or techniques used in HCI include heuristic evaluation (Nielsen, 1994b), cognitive walkthrough (Lewis & Wharton, 1997), GOMS (John & Kieras, 1996), and usability testing (Barnum, 2002).

Some have suggested that interaction patterns or pattern languages may address some of the common problems encountered in designing interactive systems by supporting reuse, capturing design knowledge, enabling the sharing of design knowledge, and facilitating communication among designers and users (Dearden & Finlay, 2006; Pemberton, 2000).

Pattern languages, in architecture and urban design, were introduced by architect Christopher Alexander and his colleagues (Alexander et al., 1977; Alexander, 1979). The intention of a pattern language, as described by Alexander (1979; C. Alexander et al., 1977), was to capture the heart of successful solutions to recurring design problems in architecture and provide a common language that architects and non-architects could both use to communicate. In the early 1990's the software engineering community caught onto the idea of using patterns to support the re-use of quality software components. The most well known example of the use of patterns in the software engineering community is Gamma, Helm, Johnson, & Vlissides (1995), commonly referred to as the Gang of Four book.

The growing interest in recent years in patterns and pattern languages in HCI is reflected in the number of workshops, panels, books, and websites dedicated to the topic (Erickson; Fincher, 2003b; Griffiths, Pemberton, Borchers, & Stork, 2000; Hillside.net; Schummer, Borchers, Thomas, & Zdun, 2004; Tidwell, 2006; van Duyne, Landay, & Hong, 2003; van Welie, Mullet, & McInerney, 2002; van Welie). Dearden and Finlay (2006) recently published a great review of this area titled *Pattern Languages in HCI: A Critical Review*.

Within the HCI literature much of the focus has been the promise of pattern languages (Pemberton, 2000), or the possible benefits they may provide. There has also been a heavy focus on the problems with using patterns in HCI, some of which include: lack of tool support, lack of a standard format, and lack of an organizing principle (Borchers, 2000; Fincher, 1999b; Seffah & Javahery, 2002; van Welie & van der Veer, 2003). Another problem discussed in the literature is the misunderstanding or misinterpretation about the difference between a pattern and a pattern language (Casaday, 1997; Dearden & Finlay, 2006; Fincher, 1999b; Mahemoff & Johnston, 2001; Todd, Kemp, & Phillips, 2004).

Although there has been a great deal of interest in this area, there is very limited empirical work to support the claimed benefits. The aim of this study is to contribute to our understanding of how a pattern language impacts the design of interfaces, at the early stages of the design process. Our results provide new insights into how patterns may be of value in design, evaluation, and communication.

1.1 Objectives/Contributions

There is little empirical evidence, in the fields of architecture (Dovey, 1990), software engineering, and HCI (Dearden & Finlay, 2006), which indicates patterns or pattern languages have a positive impact the design process or the designed product. To the best of our knowledge, there is no compelling empirical evidence which suggests that pattern languages improve the design process, or help designers create good quality designs, or designs that are of a higher quality than designs created by other means. Patterns and pattern languages are not new in HCI, yet there is a need for more empirical work in this area. In the introduction to a special design issue of *Human-Computer Interaction*, John Carroll (2006) points out that "patterns need and deserve a lot more work" (p.2). The few empirical studies which have been done within HCI including Borchers (2002), Chung, Hong, Lin, Prabaker, Landay, & Liu (2004), Dearden, Finlay, Allgar, & McManus (2002a), Dearden, Finlay, Allgar, & McManus (2002b), Finlay, Allgar, Dearden, & McManus (2002), and Saponas, Prabaker, Abowd, & Landay (2006) have conclusions that do not necessarily suggest patterns or pattern languages positively contribute to the design process or the designed product. Of these studies, only Chung et al. and Saponas et al. were able to conduct controlled experiments. Overall, the results of these studies do not provide any conclusive evidence that patterns improve the process of design or the products of design. The aim of this study is to contribute to our understanding of how patterns impact the design of interfaces, at the early stages of the design process, by empirically evaluating the impact of a pattern language on the quality of information retrieval interfaces.

1.2 Research Questions

In examining the impact of a pattern language on the design of information retrieval interfaces, the following research questions were considered.

RQ 1. Does the use of a pattern language impact the quality of designed interfaces?

In the only other controlled studies in this area, both Chung et al. (2004) and Saponas et al. (2006), examined the impact of a pattern language on the quality of the designed interfaces. Chung et al. measured quality using three variables: creativity, completeness, and quality. Saponas et al. measured quality using three similar variables: level of detail, completeness, and quality. Here, we measure quality using four variables: ease of use, level of detail, completeness, and overall quality. Ease of use was added due to its appearance in most definitions of usability (Nielsen, 2003). See Chapter 3 for more details.

RQ 2. Does the use of a pattern language impact the time it takes to design interfaces?

In both Chung et al. (2004) and Saponas et al. (2006) the authors mention that the time taken to design the interfaces was measured, but neither study reports this information.

RQ 3. How do designers view the relationships between patterns? How do designers sort and categorize the patterns?

In the HCI literature there is much discussion surrounding the difference between a pattern collection and a pattern language (Casaday, 1997; Dearden & Finlay, 2006; Fincher, 1999b;

Mahemoff & Johnston, 2001; Todd et al., 2004). In an attempt to further understand how designers view the relationships between the patterns, or the underlying language, we examine how the designers sort and categorize the patterns.

2. LITERATURE REVIEW

Human-Computer Interaction (HCI) is a multidisciplinary field, which combines the theories and practices from a number of fields including computer science, cognitive and behavioral psychology, anthropology, sociology, ergonomics, and more. HCI is defined as “a discipline concerned with the design, evaluation, and implementation of interactive computing systems for human use and with the study of the major phenomenon surrounding them” (Hewett et al., 1996). *Design* and *evaluation* are two very important words in this definition. While many, such as Fischer, McCall, Ostwald, Reeves, & Shipman, (1994), Henderson & Kyng (1991), and Karat (1997), have argued that *design* and *evaluation* are closely related, they are typically separated in practice. As seen in the definition of HCI above and in the literature, *design* and *evaluation* are dividing factors in HCI research. There is no agreed upon definition of either design or evaluation. Atwood, McCain, & Williams (2002) provide a summary of various views and taxonomies of design. Wania, Atwood, & McCain (2006) examine the interrelationships between design and evaluation specifically within HCI research.

2.1 What is this thing we call Design?

Design has been described as both an art and a science. In *Educating the Reflective Practitioner*, Donald Schön (1987) describes design as “a form of artistry” (p. 18). He goes on to say “there is no usable science of design” (Schön, 1987, p. 43). Schön describes the design process as, the reflective conversation that a designer has with his materials, or reflection in action. In *The Mythical Man Month*, Frederick Brooks Jr. (1987) describes design as “a creative process” (p. 202). In *The Timeless Way of Building*, Alexander (1979) describes that “design is often thought of as a process of synthesis, a process of putting together things, a process of combination” (p. 368). In *The Sciences of the Artificial*, Herbert

Simon (1996) describes that design “is concerned with how things ought to be, with devising artifacts to attain goals” (p. 114). Simon describes design as “changing existing situations into preferred ones” (p. 111). Here we see some similarities and difference in these views of design.

John Chris Jones (1992) reviews a number of definitions of design. Jones defines design as the “initiation of change in man-made things” (p.15). In exploring what design is, Jones describes, “The ultimate answer to the dilemma is not for designers to become as gods but for the design process to become more public so that everyone who is affected by design decisions can foresee what can be done and can influence the choices that are made” (p. 9). In a similar way Alexander (1979) suggests making the design process more transparent and participatory through the use of a pattern language. We explore this idea further in later sections. Although there is little agreement about what design actually is, there is some agreement about the challenges and complexities of design.

Challenges in Design

Some of the major challenges in design include defining the problem, deciding what exactly to design, choosing among the many design alternatives, and communicating with people from different disciplines. Design complexity has been discussed by many including Alexander (1964), Brooks (1987), Rittel & Webber (1984), and Simon (1996). Rittel & Webber contrast problems in the natural sciences, which they refer to as tame or benign problems, because they are “definable and separable and may have solutions that are findable,” with planning problems, which they describe as “wicked problems” because they lack the clarifying traits of tame problems (p. 136).

Similarly, Simon (1984b) describes design problems as ill-structured problems because they lack the characteristics of well-structured problems, some of which include that the initial problem state and the goal state may be represented in a problem space and that there is criterion for testing proposed solutions. Both Rittel & Webber (1984) and Simon acknowledge that some problems, such as designing information systems, are complex problems. Brooks (1987) echoes the same idea in stating "the hardest single part of building a software system is deciding precisely what to build" (p. 17). Brooks acknowledges that there are properties of software design, in particular, that make it different from other design tasks including: complexity, conformity, changeability, and invisibility (see Brooks, 1987 for this discussion). Alexander (1964) also acknowledges many of the same problems in design. Alexander (1964) describes,

> There are bounds to man's cognitive and creative capacity. There are limits to the difficulty of a laboratory problem which he can solve; the number of issues he can consider simultaneously; to the complexity of a decision he can handle wisely (p. 5).

Each of these authors seems to agree that these problems are inherent in design.

Communication is also seen as a problem in design, especially because design involves people from different disciplines or backgrounds working together (Kim, 1990; Rittel, 1984). Rittel (1984) describes what he refers to as "symmetry of ignorance," which refers to the idea that expertise is distributed. Thus suggesting one person cannot claim that their knowledge is superior to another person's knowledge and as a result many people should participate in design. Scott Kim (1990), in a chapter titled *Interdisciplinary Cooperation*, discusses the challenges of working with people from different disciplines. The ideas are based on studies Kim conducted at Stanford University. Kim concludes that "disciplines are like cultures: for disciplines to work well together they must learn to appreciate one another's language, traditions, and values" (p. 32). Kim presents a number of ideas about the difficulties in interdisciplinary cooperation and how to overcome these difficulties.

Although there is no agreed upon definition of design, in general, or in HCI, we seem to agree about some of the problems we encounter in design. In an attempt to address these problems we develop methods and techniques that help us cope with the problems in design. There are many well known methods and techniques used in HCI to assist in the design and evaluation of information systems. Some of the more commonly used design techniques include: task centered user interface design (Lewis & Rieman, 1994), participatory design (Kyng, 1991), user-centered design (Norman, 2002), interaction design (Preece et al., 2002), scenario-based design (Rosson & Carroll, 2002), contextual design (Beyer & Holtzblatt, 1998), paper prototyping (Snyder, 2003) and pattern languages (Dearden & Finlay, 2006). Each of these methods and techniques includes some type of structured approach which was developed to assist designers in the complex process of designing *useable* information systems. Within HCI, we all seem to agree that we would like to build systems and products which are useable and enjoyable to use. The next section is devoted to a brief discussion of usability in HCI.

Usability

The ultimate goal of design, in HCI, is to produce a product that is useable, enjoyable to use, and of course, helps a user achieve their goal, whatever it may be. But, just as there is no agreed upon definition of design there is also no agreed upon definition of usability, or how it should be measured. There are various definitions of usability (Gould & Lewis, 1985; Nielsen, 2003) and descriptions of how it may be measured (Bevan, 1995; Nielsen & Levy, 1994) in the literature (see Chapter 3 for a more detailed discussion). In an effort to design products that are useable and enjoyable, and to address the many problems that occur in design, we, as a community, have developed many design methods and techniques.

In recent years, the interest in using patterns and pattern languages to design and evaluate systems has grown. This is reflected in the number of workshops, panels, books, and websites on the topic and the increasing number of pattern collections (Dearden & Finlay, 2006; Erickson; Fincher, 2003b; Griffiths et al., 2000; Henninger & Corrêa, 2007; Hillside.net; IBM Patterns for e-business; Schummer et al., 2004; Tidwell, 2006; van Duyne et al., 2003; van Welie et al., 2002). Before reviewing Alexander's (1979; Alexander et al., 1977) work, or patterns in HCI we discuss patterns, in a general sense.

2.2 What is a pattern?

In general, the word pattern has many meanings, uses, definitions, forms, and so on. In Table 1, we present a number of common definitions of patterns in general and in HCI and software engineering. It is interesting to note that many of the definitions suggest that patterns are in some way models or forms or structures for making things. The definition from Dictionary.com suggests that a pattern can be formed naturally or by chance. Although there are similarities among these definitions, there are also differences.

Table 1. Pattern Definitions

Name	Source	Definition
Pattern	Merriam-Webster Online Dictionary	“1: a form or model proposed for imitation: exemplar 2: something designed or used as a model for making things <a dressmaker's pattern> 3: an artistic, musical, literary, or mechanical design or form 4: a natural or chance configuration”
Pattern	Dictionary.com	“1. a decorative design, as for wallpaper, china, or textile fabrics, etc. 2. decoration or ornament having such a design. 3. a natural or chance marking, configuration, or design: *patterns of frost on the window.* 4. a distinctive style, model, or form: *a new pattern of army helmet.* 5. a combination of qualities, acts, tendencies, etc., forming a consistent or characteristic arrangement: *the behavior patterns of teenagers.* 6. an original or model considered for or deserving of imitation: 7. anything fashioned or designed to serve as a model or guide for something to be made: *a paper pattern for a dress.*
Pattern	Alexander, 1979	“describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice” (p. x).
Pattern	Fowler, 1997	“an idea that has been useful in one practical context and will probably be useful in others” (p. 8)
Design pattern	Gamma et al., 1995	“are descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context” (p. 3, italics in original) “design patterns capture solutions that have been developed and evolved over time. (p. xi)
Design pattern	Holzner, 2006	“a tested solution to a standard programming problem” (p. 8)
Design pattern	Borchers, 2001	“a structured textual and graphical description of a proven solution to a recurring design problem” (p. 7)

Fowler’s definition is so general that it suggests that a pattern is simply an idea. Some definitions, such as Alexander’s, seem to emphasize problems, while others such as Holzner’s emphasize solutions. We provide a more detailed discussion surrounding definitions and forms of patterns, specifically in HCI, in a later section. The next section is devoted to Alexander et al.’s (1977) pattern language because it is one of the most cited sources for the pattern language concept.

2.3 Alexander et al.’s Pattern Language

Pattern languages in architecture and urban design were introduced by architect Christopher Alexander and his colleagues (Alexander et al., 1977; Alexander, 1979). We acknowledge that this was the work of many people, not just Alexander but from this point on we refer to the work of Alexander et al. simply as Alexander's work, for the sake of simplicity. The intention of a pattern language, as described by Alexander (1977; 1979), was to capture the heart of successful solutions to recurring design problems in architecture and provide a common language that architects and non-architects could both use to communicate. Alexander (1977) explains "Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice" (p. x).

Before Alexander and his colleagues (1977) introduced the idea of a pattern language, Alexander (1964), introduced the idea of pattern-like structures in *Notes on the Synthesis of Form*. Alexander describes these hierarchically organized pattern-like structures as a way to represent design problems which makes the problems easier to solve by reducing the gap between the designer's knowledge and the design task. Here, he draws a relationship between the fundamental elements in a pattern, the problem, the form (solution), the context, and the goodness of fit between the form and the context. What Alexander (1964) refers to, in Notes on the Synthesis of Form, as goodness of fit later evolves into the idea of *the quality without a name* (1979). We see Alexander's (1964) ideas develop further in *The Timeless Way of Building*, Alexander (1979) explains "every pattern we define must be formulated in the form of a rule which establishes a relationship between a context, a system of forces which arises in that context, and a configuration which allows these forces to resolve themselves in that context" (p. 253). In essence the patterns he is describing consist of: the context in which the problem occurs and a resolution of the forces which produces the solution in that context. From this point on we refer to the context, problem, and solution as the essence of a pattern.

Each of Alexander’s patterns contains the following elements: a name, a number, a picture (that shows an example), an introductory paragraph (which sets the context and explains how the pattern helps complete larger patterns), three diamonds (mark the beginning of the problem), headline (gives the essence of the problem in a few sentences), body of the problem (describes the empirical background), solution (describes the forces that resolve the problem in the stated context), a diagram (solution in the form of a diagram with labels to indicate its main components), three diamonds, and references to smaller patterns that are needed to complete a pattern.

Alexander’s pattern language contains 253 patterns at varying levels of scale, from patterns for cities, down to patterns for the details of a doorway. Here we describe in some detail one of Alexander’s (1977) patterns, the entrance transition pattern, seen in Figure 1. Alexander describes the problem “Buildings, especially houses, with a graceful transition between the street and the inside, are more tranquil than those which open directly off the street” (p. 549). He continues by describing the problem in more detail “The experience of entering a building influences the way you feel inside the building. If the transition is too abrupt there is no feeling or arrival, and the inside of the building fails to be an inner sanctum” (p. 549). Alexander then describes the solution to this problem by stating

> Make a transition space between the street and the front door. Bring the path which connects street and entrance through this transition space, mark it with a change of light, a change of sound, a change of direction, a change of surface, a change of level, and perhaps by gateways which make a change of enclosure, and above all with a change of view (p. 552).

112 Entrance Transition

…whatever kind of building or building complex you are making, you have a rough position for its major entrances – the gateways to the site from MAIN GATEWAYS (53); the entrances to individual buildings from FAMILY OF ENTRANCES (102), MAIN ENTRANCE (110). In every case, the entrances create a transition between the 'outside' – the public world – and some less public inner world.

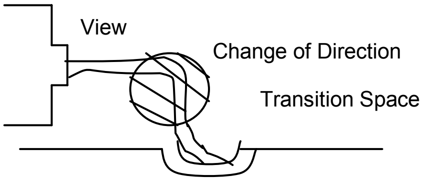
***

Buildings, and especially houses, with a graceful transition between the street and the inside, are more tranquil than those which open directly off the street.

The experience of entering a building influences the way you feel inside the building. If the transition is too abrupt there is no feeling or arrival, and the inside of the building fails to be an inner sanctum.

The following argument may help to explain it. While people are on the street, they adopt a style of 'street behavior.' When they come into a house they naturally want to get rid of this street behavior and settle down completely into the more intimate spirit appropriate to a house. But it seems likely that they cannot do this unless there is a transition from one to the other which helps them to lose the street

Therefore: **Make a transition space between the street and the front door. Bring the path which connects the street and entrance through this transition space, and mark it with a change of light, a change of sound, a change of direction, a change of surface, a change of level, perhaps by gateways which make a change of enclosure, and above all a change of view.**



***

Emphasize the momentary view which marks the transition by a glimpse of a distant place – ZEN VIEW (134); perhaps make a gateway or a simple garden gate to mark the entrance – GARDEN WALL (173); and emphasize the change of light – TAPESTRY OF LIGHT AND DARK (135)

Figure 1. Portions of an Entrance Transition Pattern adapted from Alexander et al. (1977, pp. 548-552)

The patterns which make up Alexander's (1977) pattern language contain relationships to one another, as described in the introductory paragraph and the concluding reference paragraph of each pattern. This is not always the case in HCI pattern collections. Alexander's pattern language is not just a collection or set of patterns. Instead the patterns, within Alexander's pattern language, are related to one another, in that higher level patterns are made up of lower level patterns, and these relationships are made explicit within the patterns. Alexander (1979) describes "the structure of the language is created by the network of connections among individual patterns: and the language lives, or not, as a totality, to the degree these patterns form a whole" (p. 305). Alexander (1979) describes the relationships between the patterns and the network in which the patterns exist by stating

> Each pattern sits at the center of a network of connections which connect it to certain other patterns that help to complete it…and it is the network of these connections between patterns which creates the language…In this network, the links between the patterns are almost as much a part of the language as the patterns themselves (pp. 313-314).

Here we see that Alexander distinguishes a pattern language from other things by the network of relationships between the patterns, the smaller patterns contained in the larger patterns, and their ability to form a whole. These ideas will be discussed in later sections in the context of HCI.

The Quality without a Name

The aim of Alexander's (1977) pattern language is to build things which have what he refers to first as *goodness of fit* (1964) and later as *the quality without a name* (1979). Alexander (1979) explains "there is a central quality which is the root criterion of life and spirit in a man, a town, a building, a wilderness. This quality is objective and precise, but it cannot be named" (p. 19). He continues by stating "the fact that this quality cannot be named does not mean that it is vague or imprecise. It is impossible to name because it is unerringly precise…words fail to capture it because it is much more precise than any word" (p. 29). Alexander points to a number of words which are used to talk about the quality without a name including: alive, whole, comfortable, free, exact, egoless, and eternal. Although Alexander does not identify one word to describe the quality without a name, he points out that each word captures some of what the quality is. In addition, he describes why each word falls short of describing the quality. The goal of documenting the patterns is to capture and describe the quality without a name so that others can understand the quality and the context in which it exists.

Alexander (1979) proposes that anyone, not only designers, can use a pattern language to design buildings. He also suggests that a pattern language can be used as a common language, or a *lingua franca* (Erickson, 2000), for both designers and inhabitants of the buildings. Alexander (1979) describes “In this same way, groups of people can conceive their larger public buildings, on the ground, by following a common pattern language, almost as if they had a single mind” (p. 427). Alexander suggests a pattern language may serve as a type of boundary object (Carlile, 2004; Star, 1990) which may enable communication among people from different disciplines.

Alexander’s work began with one book describing pattern-like structures. Over the years this work has evolved into multiple books and multiple volumes (see http://www.patternlanguage.com/). Alexander’s ideas have clearly influenced people in many fields but, the empirical validation of his work has only just begun.

The Use of Pattern Languages in Architecture

There are few examples of pattern languages being used in architecture, outside of Alexander’s (1975; 1981) work. Most of the claims made about pattern languages being used seem to be based solely on the fact that *A Pattern Language* is a best selling book (Erickson, 2000). Erickson (2000) points out that there are some published accounts of architecture projects using patterns and he gives one reference (Fromm & Bosselmann, 1984) which summarizes a few of them. Borchers (2001) in the preface to *A Pattern Approach to Interaction Design* states “*pattern languages* have been communicating design knowledge successfully in architecture and software engineering in the past” (italics in original) (p. xi).

Although there are many claims made about the successful use of pattern languages there are few concrete examples and empirical studies that can be found to support these claims.

2.4 Patterns in other disciplines

Patterns and pattern languages have been documented in a variety of disciplines including architecture, software engineering (Gamma et al., 1995), pedagogy (Fincher, 1999a), e-business (IBM Patterns for e-business), and HCI (Dearden & Finlay, 2006). In the following sections we briefly summarize the work that has been done outside of HCI and then focus mainly on work that has been done with the field of HCI.

Patterns in Software Engineering

In the early 1990's the software community caught onto the idea of using patterns to support the re-use of quality software components. The most well known example of the use of patterns in the software community is the Gang of Four (Gamma et al., 1995). Patterns have been used in software engineering (SE), but not necessarily in the way Alexander intended. Dearden, Finlay, Allgar, & McManus (2002b) point out that the SE approach to patterns ignores the participatory aspects of Alexander's PL and that the SE community has placed more of an emphasis on knowledge sharing between professionals rather than between professional and users.

2.5 From Architecture to HCI: Promises, Promises

In HCI, *the promises of pattern languages* (Pemberton, 2000) make the use of pattern languages very attractive. The promises of pattern languages are seen in the many possible benefits of using patterns and pattern languages. It has been suggested that patterns may support reuse and may be used as a *lingua franca* (Erickson, 2000), or a language for design which may help designers and non-designers communicate. It has been suggested that patterns may also help capture design knowledge, in addition to supporting the sharing of this knowledge. In addition, it has been suggested that patterns and pattern languages may be able to serve as both design and evaluation tools. The promises have been an interesting area of discussion in HCI.

Fincher et al. (2003) point out that there have been at least two motivations for exploring patterns and pattern languages in HCI,

> It is relatively easy to make an analogy between the domains of architecture and UI design, based on concern for the effect of a constructed artifact on personal and social behaviours…Alexander's patterns (the "first encounter" with patterns for most) "make sense" to designers. They are also written compellingly and elegantly (p. 1044).

Fincher et al. explain a number of similarities between architecture and interaction design, one similarity being that in both disciplines a designer designs a space or an artifact for use by someone else. In architecture this may be a room in a building or a building itself. In interaction design this may be a website or an application for a handheld device. In both cases the user interacts with the artifact, which over time may change based on the way the users interact with it. Fincher et al. also point to the fact that Alexander’s patterns seem to “make sense” to many of us. It seems that they make sense during our first encounter because the patterns seem like reasonable solutions to common design problems. Another reason they may just “make sense” to us could be because we have actually seen some of the patterns in our everyday experiences.

Erickson (2000) points to the two most often cited reasons for the use of pattern languages: quality and reuse. In both architecture and HCI, and probably every other design discipline, we strive to design quality artifacts that people will use and enjoy, without having to reinvent the wheel every time. Beyond just quality and reuse Erickson describes a number of things which make Alexander’s pattern language suitable for generating a *lingua franca* or a common design language. As described earlier, communication is seen as a major problem in design, therefore the HCI community is very interested in trying to reap the benefits of something that promises to provide a common design language.

Granlund, Lafreniere, & Carr (2001) point out that user interface designers are faced with problems which have generally known solutions but there is often a problem communicating the solutions to others. Granlund, Lafreniere, & Carr acknowledge guidelines as a possible solution to this problem but, they point out that guidelines are seen as hard to interpret and that finding such guidelines may require considerable effort (Mahemoff & Johnston, 1998b). Granlund, Lafreniere, & Carr see reuse as one reason why there has been increasing interest in patterns and patterns languages in HCI. They also acknowledge many of the reasons, mentioned earlier, for interest in the topic including that patterns may offer a way to capture and transfer knowledge, patterns may provide a lingua franca, and patterns may support both analysis and design (Granlund, Lafreniere, & Carr 2001).

Design is a complex task that involves people from many different disciplines. In an earlier section we reviewed some of the major problems in design including design complexity and communication in design. As suggested by many within HCI, patterns aim to solve some of these problems in design. Although there is a great hype about the promise of pattern languages in HCI (Pemberton, 2000), there is little empirical work to support these claims (Dearden & Finlay, 2006). Being that there have been very few empirical studies in this area, we know very little about how patterns may be of value in HCI.

2.6 Patterns and Pattern Languages in HCI

Despite the more recent interest in the topic, there were references to Alexander’s work prior to the first workshop in 1997. In *User Centered Systems Design: New Perspectives on Human Computer Interaction*, Norman and Draper (1986) refer to Alexander’ work and Norman (1998) also refers to his work in *The Design of Everyday*

Things. Rheinfrank, Hartman & Wasserman (1992) and Rheinfrank & Evenson (1996) described something very similar to a pattern language, but they refer to the language as a design language.

Just as the literature focuses on the promise of pattern languages, the literature also focuses on the problems with using patterns in HCI including: lack of tool support, lack of a standard format, and lack of an organizing principle (Borchers, 2000; Fincher, 1999b; Seffah & Javahery, 2002; van Welie & van der Veer, 2003). Another problem discussed in the literature is the misunderstanding or misinterpretation about the difference between a pattern and a pattern language (Casaday, 1997; Dearden & Finlay, 2006; Fincher, 1999b; Mahemoff & Johnston, 2001; Todd et al., 2004). Despite all that has been written about patterns and pattern languages in HCI, there is little empirical evidence that suggests pattern languages somehow aid in creating quality designs or designs of higher quality than those produced by other techniques or methods.

Before discussing the empirical work relating to patterns in HCI we summarize various definitions of patterns and pattern languages in HCI and describe a number of research issues which have been addressed and/or recognized by the HCI community.

What is a pattern in HCI?

In an earlier section we reviewed a number of general pattern definitions. Here, we present various definitions of patterns specifically in HCI. In addition, we discuss the elements that together make up patterns in HCI. As seen in Table 2, within HCI, there is no agreed upon definition of a pattern or the form a pattern should take. Tidwell (2006) describes

> In essence, patterns are structural and behavioral features that improve the 'habitability' of something – a user interface, a web site, an object-oriented program, or even a building. They make things easier to understand or more beautiful; they make tools more useful and usable…patterns can be descriptions of best practices within a given design domain. They capture common solutions to design tensions (usually called "forces" in pattern literature) and thus, by definition, are not novel (p. xiv).

We would like to point out that Alexander (1977) in his definition of a pattern states "each pattern describes a problem which occurs over and over again in our environment" (p. x). This is a very important distinction in the definition of a pattern that is not consistent across all definitions in HCI, see Table 2. Borchers (2000) points out that a pattern is a *proven* solution to a recurring design problem (italics added for emphasis) and Tidwell (2006) points out that patterns are not novel. These authors emphasize that patterns are things that exist in the world. Both Borchers and Tidwell have published a pattern language, and each one points out that the patterns in their pattern language are patterns that they have been in contact with in the real world. In HCI, it is not always the case that the authors have been in contact with the patterns they document. See discussion of pre-patterns in Chung et al. (2004) and Saponas et al. (2006).

Table 2. Patterns in HCI

Author	Pattern Definition
Borchers, 2000	"a proven solution to a recurring design problem" (p. 369)
Tidwell, 2006	"capture common solutions to design tensions (usually called forces in pattern literature) and thus by definition are not novel…this book describes patterns literally as solutions to design problems because part of their value lies in the way they resolve tensions between various design contexts" (p. xiv)
Granlund et al., 2001	"a pattern is a format for describing a solution to a design problem" (p. 1)
van Duyne et al., 2003	"Patterns communicate insights into design problems, capturing the essence of the problems and their solutions in a compact form. They describe the problem in depth, the rationale for the solution, how to apply the solution, and some of the trade-offs in applying the solution" (p. 19)

Each of the definitions above includes a reference to a problem and a solution, and most, but not all, definitions include a reference to the context. In these definitions we see agreement with what we refer to as the essence of a pattern, the problem, context and solution.

Dearden and Finlay (2006) provide a table that summarizes different perspectives on the essential characteristics of a pattern. This table helps identify the points of contention within HCI regarding what a pattern is. Some of the more interesting characteristics include:

- A pattern is a part of a language
- A pattern is validated by use
- A pattern includes its rationale
- A pattern is grounded in a domain
- Patterns support a ‘lingua franca’
- Patterns reflect design values
- Patterns capture design practice
- Different patterns deal with problems at different ‘scales’

Dearden and Finlay (2006) point to a general agreement within the HCI community that patterns should allow communication between different groups, patterns languages as opposed to single patterns are important, patterns address problems at different levels, and patterns involve questions of value. These characteristics seem to be points of contention in HCI mostly because we have a lack of empirical work to base any of these claims on. Those that we find of particular interest include whether a pattern needs to be part of a pattern language or whether a collection of patterns, that may not contain relationships between patterns, is also useful. There is also disagreement within the community about whether patterns should be validated by use or if there is some value to publishing patterns before they have been validated or exist in the environment. It is also not clear whether patterns need to be grounded in a specific domain to be useful. Some of these issues will be addressed in more detail in later sections. Although there may be general agreement on some of these things within the HCI community, we have very little empirical evidence to support any of these claims.

Pattern Presentation

In HCI literature patterns are described as comprising of various elements. Most patterns in HCI include some variation of the elements in Alexander's patterns, described earlier. Table 3 compares the elements included in patterns by various authors in HCI. Casaday (1997) uses what he refers to as the main elements in Alexander's original definition of a pattern: name, context, forces, problem, and solution. Dearden, Finlay, Allgar, & McManus (2002b) and Borchers (2001) follow a style similar to Alexander's (1977).

All authors in Table 3 include a name or title for a pattern, in addition to a solution. A list of pattern collections, and languages, and their websites can be found in Appendix A. Most authors include a diagram, context, and problem. About half of the authors in the table include a reference section that describes how a pattern is related to other patterns. It seems that there is some agreement in the community that patterns contain what we refer to as the essence of a pattern: problem, context, solution.

Table 3. Elements in HCI patterns

Tidwell (2006)	Van Duyne et al. (2003)	van Welie	Yahoo!	Casaday (1997)	Borchers et al. (2001)	Dearden et al. (2002b)	Chung et al. (2004)
Name	Name	Title	Title		Title	Title	Title
	Background						Background
What	Problem	Problem	Problem Summary	Problem	Problem Statement	Problem	Problem
Example	Example	Example	Example		Existing Examples	Illustration	Example
Why		Why	Rationale	Forces			
Use When		Use When	Use When	Context			
	Solution	Solution	Solution	Solution	General Solution	Solution	Solution
How		How					
Diagram		Diagram				Diagram	
	Consider these other parts				References	References	
		Implementation					
		Literature					References (to literature)
			Accessibility (not in all patterns)				

The differences in the formatting of the patterns and the elements they contain do not matter much if these remain as separate collections but, if we, as a community, want to be able to grow these collections and languages together and provide tool support for finding patterns we may need to come to an agreement on the elements of a pattern and the format it should take. Deng, Kemp, & Todd (2005), Fincher et al. (2003), and Henninger & Ashokkumar (2005) have suggested using tools or languages to access and format patterns. In the next section we will discuss the similarities and differences between patterns and other HCI design and evaluation techniques, specifically guidelines.

Patterns and Guidelines in HCI

There has been much discussion in HCI literature about the similarities and differences between patterns, guidelines, claims, and other design techniques. Many comparisons have discussed patterns and guidelines (Griffiths & Pemberton, 2005; Henninger & Ashokkumar, 2005; Kotze, Renaud, Koukouletsos, Khazaei, & Dearden, 2006). Dearden and Finlay (2006) identify a number of similarities and differences between patterns and guidelines including:

- Patterns use more specific examples
- Patterns include a statement of the “problem” they address
- Patterns deliberately scope the context of application
- Patterns explicitly reflect particular design values

In contrasting patterns and guidelines Kotze et al. (2006) point out that guidelines do not include their rationale, may leave out examples, and are generally independent of the rest of the guidelines within a set of guidelines. Henninger and Ashokkumar (2005) identify perspective and representation of information as one main difference between patterns and guidelines. They argue that usability patterns are more problem oriented, describing a problem and a solution, while guidelines provide more general information or advice (Henninger & Ashokkumar, 2005). Dearden and Finlay (2006) point out patterns are derived from practice rather than theory.

In their discussion of the differences between patterns and guidelines Granlund et al. (2001) point to the fact that the user, the task, and the context are all missing from guidelines, in addition to design rationale. It seems that many of the distinctions between patterns and other techniques address that patterns include examples, the context in which the problem occurs, and that the relationships between patterns are explicit in a pattern language. One of the more prominent distinguishing factors is that the patterns in a pattern language are related to one another and that the network of relationships is revealed in the pattern language. Guidelines may also be related to one another but the relationships are not always explicit. In the next section we distinguish a pattern from a pattern language and further define a pattern language.

What is a Pattern Language?

In the HCI community, there seems to be a misunderstanding or misinterpretation about the difference between a pattern and a pattern language. We would like to make a distinction between a pattern and a pattern language. It is clear, in reading the literature, that not everyone recognizes the difference between a pattern and a pattern language. Mahemoff & Johnston (2001) point out that “it is often difficult to see how a pattern collection could offer true benefits to a practitioner,” they continue by saying “our view is that the critical notion of “language” in “pattern language” is all too often overlooked” (p. 350). We tend to agree with Mahemoff & Johnston and therefore, would like to make a clear distinction between a pattern and a pattern language.

When referring to a pattern, we are referring to a single pattern, as described by Alexander (1977; 1979), Tidwell (2006), Borchers (2000), or Mahemoff & Johnston (2001). A pattern describes a problem which occurs over and over again in some context, in addition to describing a solution to the problem. When referring to a pattern language we are not simply referring to multiple patterns. In order for a collection of patterns to be a pattern language there should be an organizing principle, that is, the patterns in the language should to be connected by meaningful relationships. The patterns in the language must also include information about situations that are appropriate for application of the specific pattern and other related patterns. In addition to an organizing principle a pattern language must contain patterns at different levels of scale. Therefore in order to be a pattern language there must be an organizing principle, explicit relationships between the patterns, in addition to patterns at different levels of scale.

Fincher and Windsor (2000) suggest some requirements for an organizing principle including:

- It should allow users to find patterns
- It should enable users to find related patterns
- It should allow users to evaluate the problems from multiple viewpoints
- It should allow users to build new solutions

Table 4 contains various definitions of pattern languages. Some definitions are very vague while others are quite detailed. Mahemoff & Johnston (2001) point out that Alexander's pattern language begins with the distribution of towns and moves down to finish at the level of detailed construction. Here, the idea of a network appears, rather than just a hierarchy. In addition, they point out that lower level or more detailed patterns help to complete the higher level patterns. Mahemoff & Johnston also recognize that pattern languages are not objective, but rather subjective. This echoes the idea that patterns represent or reflect values, as mentioned earlier.

Table 4. Pattern Language Definitions

Author	Pattern Language Definition
Mahemoff & Johnston, 2001	"a *pattern language* is formed when a collection of patterns is arranged into a network of interdependent patterns, especially where higher-level patterns yield contexts which are resolved by more detailed patterns" (p. 351)
Wesson & Cowley, 2003	"a complete set of related patterns" (p. 1)
Todd et al., 2004	"collections of related patterns which are organised and linked into one or more interlocking hierarchies" (p. 91)

Todd, Kemp & Phillips (2004) review several definitions of pattern languages and note that all the definitions indicate “that the linking between patterns on one level can form a higher-level pattern that includes information not available from the individual patterns alone” (p. 92). Salingaros (2000) points out that “a loose collection of patterns is not a system, because it lacks connections” (p.154). Granlund et al. (2001) state that "patterns must also be part of a language of interrelated patterns, participating in and supporting each other, in order to be truly useful" (p. 2).

Although not everyone in the HCI community recognizes the difference between a pattern and a pattern language there seems to be a group of authors that agree that a pattern language should contain patterns that have relationships to one another and patterns at different levels of scale. In the next section we explore the structure and organization, or relationships between patterns, in a pattern language.

Pattern Languages: Structure and Organization

Alexander’s (1977) pattern language consists of patterns at different levels of scale. It contains higher level patterns for regions, including: the distribution of towns and country towns. It also contains lower level patterns for use within the communities in those regions, for example, parallel roads, row houses, local town hall, pools and streams, street café, and beer hall. It contains even lower level patterns for the buildings on the land in the communities, including: shielded parking, main entrance, sheltering roof, alcoves, half-open wall, waist-high selves, and windows which open wide. Here it is easy to see how the lower level patterns may be used to help complete the higher level patterns.

In describing how to use a pattern language Alexander refers to the network of the language and the different scales at which patterns exist. Alexander (1977) describes,

> When we use the network of a language, we always use it as a sequence, going through the patterns, moving always from the larger patterns to the smaller, always from the one which create structures, to the ones which embellish those structures, and then to those which embellish the embellishments (p. xviii).

In describing the relationships and connections between the patterns at different levels of scale Alexander (1979) states,

> Each pattern then, depends both on the smaller patterns it contains, and on the larger patterns within which it is contained…Each pattern sits at the center of a network of connections which connect it to certain other patterns that help to complete it (pp. 312-314).

Again, Alexander stresses the importance of the connections between patterns by stating "In this network, the links between the patterns are almost as much a part of the language as the patterns themselves" (pp. 312- 314). Here, he explicitly states the importance of the relationships between the patterns.

The need for different levels of scale and an organizing principle has been mentioned in HCI literature but, not fully addressed. There is no agreement about how patterns should be organized or how the relationships between them should be structured, although there have been some suggestions. Todd, Kemp & Phillips (2004) point out that Alexander's language is ordered by a hierarchical scale from a city down to artifacts such as individual doors. They also point out that Alexander's language has only one root, Pattern 1, Independent Regions. Having only one root is also something that is not always seen in pattern languages in HCI.

van Welie & van der Veer (2003) suggest possible ways of organizing patterns including: by function, by problem similarity, by user task, and by user type. van Welie & van der Veer present a hierarchical partial pattern language for web design which contains a number of different levels including: posture level, experience level, task level, and action level. Figure 2 depicts van Welie & van der Veer's suggested levels with some example patterns from their partial pattern language for web design. The posture level is the purpose or reason for existence, the experience level contains the main user goals and tasks. The task level describes a series of interactions on objects and the action level is the lowest level they consider a pattern.

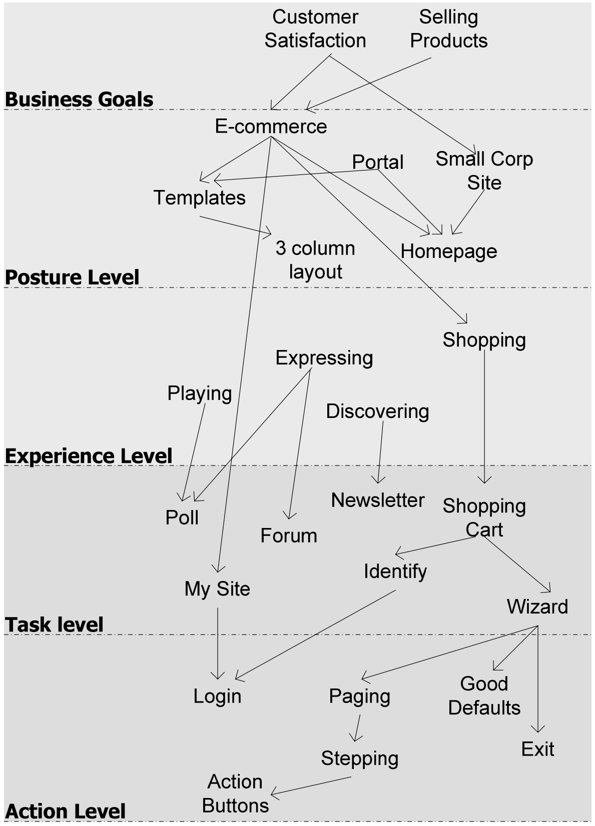


Figure 2. Pattern Language Levels (adapted from van Welie & van der Veer, 2003)

Borchers et al. (2001) suggest that for HCI design patterns the most natural organizing principle is Alexander's notion of scale. Borchers et al. created a list of categories that HCI design patterns should fall into. The categories include: society (beyond systems), multiple users, social position, system, application, UI structure (dialogue), components (containers, windows, layout), primitives (buttons and other simple widgets), and physical properties. Borchers et al. also identify a second fundamental organizing principle as levels in the HCI design process, from analysis oriented to structure oriented patterns. As seen in Figure 3 they include: culture and society, environment, role, use, navigation, and tasks within these two levels.

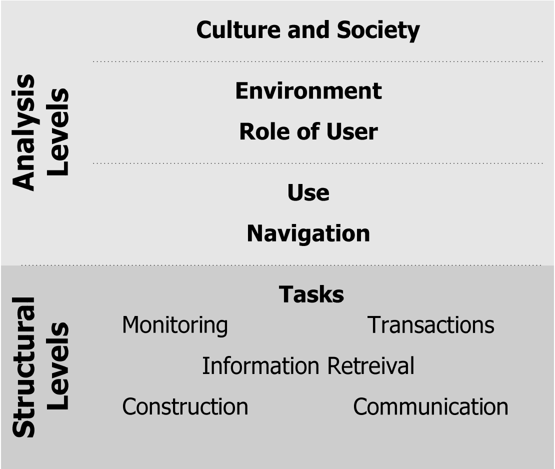


Figure 3. An Organizing Principle (adapted from Borchers et al., 2001)

Henninger & Ashokkumar (2005) suggest an ontology based structure for organizing pattern languages. The relationships between the patterns are based on the class, subclass structure and there are patterns at levels similar to those identified by van Welie & van der Veer (2003) including experience level, task level, and action level. Henninger & Ashokkumar suggest relationships between patterns which include: contains, is equivalent, is an alternative, is specialization, is to be used in combination with, and is disjoint with. Henninger & Ashokkumar stress that “too many pattern collections have been created that draw little or no relationships between each other” (p.52). They continue by saying they see context as one of the main organizing features of patterns. This focus on context is also reflected in the way they suggest pattern languages that are domain specific, such as their example from online shopping.

Many authors have suggested specific organizing principles while others such as Borchers, (2000) stress that there may be more than one appropriate organizing principle for pattern languages within HCI. From our limited empirical evidence, it is not yet clear if the patterns themselves, or the patterns and the organizing principle together as a pattern language, may provide more benefit in design and evaluation in HCI.

A Language for Design

Erickson (2000) suggested using pattern languages as a *lingua franca*, or a common language which allows designers and non-designers to communicate with one another. As described above, Alexander intended his pattern language to be used by both designers and non-designers. As designers, we know communication is a very important part of design, yet it can be a source of problems when people from different disciplines try to communicate with one another (Kim, 1990; Rittel & Webber, 1984).

Erickson (2000) describes design as a communicative process and he points to the tools we use in design that aim to improve communication as evidence, for example, storytelling, scenario-making, prototype building, and user testing. Erickson describes a lingua franca “as a common language which is accessible to all participants in the design process” and argues that the development of a *lingua franca* should be part of the process of interactive systems design (for each project) (2000, p. 358).

Bayle et al. (1998) report on the results of a workshop at CHI 97 which focused on the use of patterns and pattern languages in interaction design. Bayle et al. point to some properties of pattern languages that might enable them to serve as lingua franca for designers in HCI including pattern languages:

- are based on concrete prototypes
- work at multiple levels (community, group, individual) and they try to tie the levels together
- attempt to bridge the gap between the physical and social worlds
- are amenable to gradual development (p. 18)

Although we have little or no empirical support for these claims, there is hope that pattern languages may be able to provide a lingua franca in HCI. In the next section we explore how linguistic theory may help us understand the “language” aspect of pattern languages.

2.7 Understanding the “Language” Aspect

Within the HCI community there is a misunderstanding or misinterpretation about the difference between a pattern and a pattern language. In an attempt to further explore the “language” aspect we look to linguistic theory. This community has already spent years trying to understand language therefore this seems like a field we can learn from. We look broadly at linguistics and more specifically at Chomsky’s phrase structure grammar or universal grammar.

Language is defined by Merriam-Webster’s online dictionary as:

> 1 a: the words, their pronunciation, and the methods of combining them used and understood by a community b (1)**:** audible, articulate, meaningful sound as produced by the action of the vocal organs (2)**:** a systematic means of communicating ideas or feelings by the use of conventionalized signs, sounds, gestures, or marks having understood meanings…(5)**:** a formal system of signs and symbols (as FORTRAN or a calculus in logic) including rules for the formation and transformation of admissible expressions.

Linguistics is the scientific study of language (Trask, 1999). As described by Crystal (1997b), Charles Hockett introduced 13 design principles which he used to distinguish spoken languages from other modes of communication. If this list is limited to those principles that apply to languages in general, and not specifically spoken languages, the list is reduced to 3 properties.

- A language must have an agreed upon **set of symbols**
- The meaning of the symbols is **arbitrary** and there are no clear relationships between the symbols
- A language must contain a **grammar**, or a set of rules to manipulate the symbols and these rules can be used to create an infinite number of utterances

Pattern languages in HCI may exhibit these three properties, but not all the 13 properties of a language. Using these 3 properties we can examine whether pattern languages exhibit the same properties. Pattern languages, as described in HCI and by Alexander, seem to have a set of symbols, in Alexander's case it was 253 patterns or the names of the patterns. It also seems that the meaning of the symbols is arbitrary, in the sense that the name could be changed to something other than what it already is. In looking at Alexander's pattern language, it seems to contain a grammar or a set of rules for combining the patterns. In Alexander's pattern language this is expressed in the introductory paragraph, which sets the context, and in the concluding paragraph which ties the patterns to the smaller patterns which complete the pattern. In HCI, many times there is a section for related patterns or references, as seen in Table 3, but, it seems that there is not always a set of rules for combining the patterns. In some cases, for example, in van Duyne et al.'s (2003) pattern language there is a set of rules for combining the patterns, but in other cases, like van Welie's patterns there is not an explicit set of rules for combining the patterns.

In linguistics there are many models of language that attempt to explain how it is organized. One of the more common models of language is the three level model of language, seen in Figure 4. This model takes grammar to represent one branch of language structure, distinct from pronunciation (phonology) and semantics (meaning) (Crystal, 1997a).

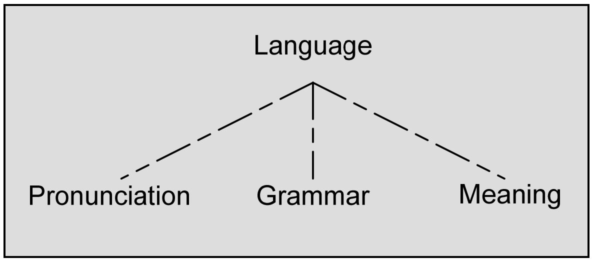


Figure 4. Three Level Model of Language Structure (adapted from (Crystal, 1997a)

Chomsky (1957) introduced the more specific notion of syntax and the general notion of grammar, as seen in Figure 5. Chomsky (1957; 1975) proposed that a theory of language must describe the complete set of **rules** which will generate all the sentences of the language and give an account of the underlying **syntactic structure** of sentences.

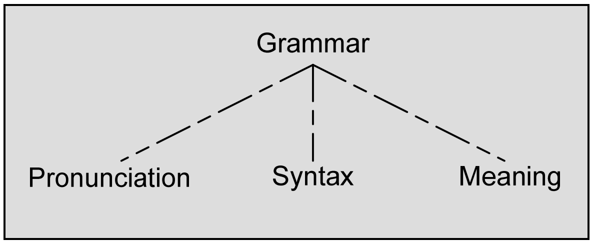


Figure 5. Chomsky’s General Sense of Grammar (adapted from Crystal, 1997a)

In comparison, Alexander's pattern language contained structure, organization, and patterns with different levels of scale. Alexander also stressed that the connections between the patterns are more important than the patterns themselves. As already stated, this is not always the case with patterns in HCI. We suggest that a pattern language must contain rules for combining patterns in the form of an **organizing principle** or explicit **relationships** between patterns. In addition, a pattern language should have an underlying structure that provides an order for applying patterns at different **levels of scale**. Because Alexander refers to *A Pattern Language* as a "language" we also refer to work inspired by his ideas as pattern languages, yet we are not suggesting that these things are true "languages" as described by Crystal (1997) in *The Cambridge Encyclopedia of Language*.

The idea of a pattern language, as described by Alexander, resembles the idea of a grammar more than it resembles the idea of a "language." There is a large body of literature surrounding grammars. There are grammars for describing and generating artifacts such as coffeemakers (Agarwal & Cagan, 1998) and architecture, from different areas of the world (Chiou & Krishnamurti, 1995; Downing & Flemming, 1981). There are also interactive systems that allow users to generate grammar based designs (Chase, 2002). This is another area that we may want to further investigate in the future.

In linguistics, there are clear properties that distinguish a language from something that is not. In HCI we do not seem to have clear rules for distinguishing something that is a pattern language from something that is not. We suggest that we use the 3 properties of a language and Chomsky's theory of language as a starting point for defining what a pattern language is in HCI. We would again like to point out that we do not fully understand the value a pattern language provides that a pattern collection does not. We need more empirical work in this area. It is also not clear whether linguistic theory influenced Alexander's work but it is easy to see the similarities. In the next section we discuss Alexander's process for the discovery of a pattern language.

2.8 Constructing a Pattern Language/Observing Patterns

Alexander (1979) describes the process of constructing a pattern language as a process in which patterns are observed and discovered from the built environment. In HCI, it is not always the case that the patterns have been observed and discovered from existing systems. In some cases, the patterns have been developed with very little observation (Chung et al., 2004; Saponas et al., 2006). In the following section we summarize Alexander's process for discovering patterns and constructing a pattern language, in addition to other processes which have described in HCI literature.

According to Alexander (1979) "In order to discover patterns which are alive we must always start with observation" (p.254). Alexander describes the process through which patterns can be discovered through observation,

> Try to discover some property which is common to all the ones which feel good, and missing from all the ones which don't feel good…This property will be a highly complex relationship…Now try to identify the problem which exists in entrances which lack this property…Knowledge of the problem then helps shed light on the invariant which solves the problem…Sometimes we find our way to this invariant by starting with a positive set of examples…At other times, we may discover the invariant by starting with the negative examples, and resolving them (pp. 255-258).

Alexander points out that patterns may be discovered in different ways, by identifying the problem and later finding a solution, or by seeing a positive set of examples and therefore recognizing a solution. Alexander describes how this is a process of discovery, it is "a discovery in the sense that it is a discovery of a relationship between the context, forces, and relationships in space" (p. 259). It is important to note that we do not know whether or not patterns need to be discovered in order to be useful. At this time there is no evidence to support either argument.

In HCI, it is not always the case that patterns have been observed and discovered from existing systems. In some cases, pattern authors are simply documenting good design they encountered during their many years of experience (Tidwell, 2006; van Duyne et al., 2003). In van Welie's terms "I try to capture every bit of good design that I encounter." In some cases, such as Wellhausen (2006), the authors of the pattern collections do not explicitly mention where their patterns come from. And yet in other cases, such as Chung et al. (2004) and Saponas et al. (2006), the patterns have been developed by the authors without existence of such systems. Others such as Di Lucca, Fasolino, & Tramontana (2005) have explored automatic identification of patterns based on characteristic features.

Whether the patterns in a pattern language have been discovered or simply described naturally leads to a discussion of whether or not patterns and pattern languages really are *patterns* or *pattern languages*. We can use linguistic theory as described above as a starting point, but there has also been discussion of other ways of validating pattern and pattern languages in the literature.

2.9 Validating a Pattern or a Pattern Language

In HCI there has been some discussion of whether or not to validate patterns and pattern languages in some way in order to ensure that the patterns being documented are real patterns (Lafreniere & Hedenskog, 2001; Todd et al., 2004; Winn & Calder, 2002). Alexander's (1979) validation consists of fulfilling two conditions. He describes,

> We say that a pattern is good, whenever we can show that it meets the following two empirical conditions: 1. The problem is real. This means that we can express the problem as a conflict among forces which really do occur within the stated context, and cannot normally be resolved within the context. This is an empirical question. 2. The configuration solves the problem. This means that when the stated arrangement of parts is present in the stated context, the conflict can be resolved, without any side effects (pp. 282-283).

Some authors in HCI point to rules for identifying whether or not something is a pattern. For example, Lafreniere & Hedenskog (2001) mention the rule of three. In other words, if you can find three examples of a pattern it may be a pattern. Winn & Calder (2002) propose a list of nine essential characteristics of software patterns which can be used to determine whether something may or may not be a pattern. According to Winn & Calder a pattern:

- Implies an artifact
- Bridges many levels of abstraction
- Is both functional and nonfunctional
- Is manifest in a solution
- Captures system hot spots
- Is part of a language
- Is validated by use
- Is grounded in a domain
- Captures a big idea (pp. 60-65)

Of these nine characteristics there are some that are not found in all patterns in HCI, for example, patterns are not always part of a language, they are not always validated by use, and they are not always grounded in a domain. Although this list of nine characteristics is interesting we do not have any empirical support indicating these are the nine characteristics of a pattern.

Todd, Kemp & Phillips (2004) refer to Salingaros' (2000) two forms of connectivity for validating a pattern language: external validity and internal validity. According to Salingaros external validity examines the "feel right" factor or value system while internal validity examines the organizing principle(s) and the connectivity between the levels of the language in order to determine whether higher level patterns are made of lower level patterns.

The participants of a CHI 2002 workshop discussed other terms suitable for evaluating a pattern language's external validity including: Breadth, Depth, Applicability, Clarity, and Convenience (Todd et al., 2004). Todd et al. propose six tests or questions which can be applied to a pattern language to determine its internal validity.

> Test 1 – Do the reference and context links between the patterns form a map?
> Test 2 – Does the context map match the reference map?
> Test 3 – Can the map be ordered into a hierarchy of levels?
> Test 4 – Can the levels be used to describe a user interface at different degrees of granularity (scale)?
> Test 5 – How 'rich' are the links within each level of the hierarchy?
> Test 6 – Can the patterns be organized by different classification systems thereby providing alternative viewpoints? (p. 93)

Todd et al. used these questions to evaluate three existing collections of patterns: van Welie – GUI collection, van Welie – WEB collection, and Borchers – HCI collection. None of the collections passed all six tests (see Todd et al., 2004 for the results). After the collections did not pass the test they argued for relaxing the tests which would allow for the Borchers – HCI collection to be considered a pattern language. Although these six questions seem valid, again we do not have any empirical support that indicates that these six questions validate something as a pattern language.

Bayle et al. (1998) point out that it is difficult to make good patterns and that they are not sure how to build an entire pattern language. They acknowledge that it took Alexander and his colleagues a decade and they recognize that building a pattern language needs to be a collaborative effort. The authors also point out that creating patterns is an exercise in applying values, in other words what people think is of value or important will be displayed in a pattern.

There is obviously no agreement on what makes a pattern, a pattern, or what makes a pattern language, a pattern language. This is an area that clearly needs more attention and more empirical work. In the next section we provide a brief overview of some of the pattern collections in HCI.

2.10 Patterns and Pattern Languages in HCI Literature

A number of HCI patterns and pattern collections or pattern languages exist today. Henninger & Corrêa (2007) point to more than 400 published patterns in over a dozen collections within HCI. We provide a brief overview of the patterns described by Borchers (2001), Tidwell (2006), van Duyne et al. (2003), van Welie, Wellhausen (2006), and Yahoo!. We also recognize the work on design languages described by Rheinfrank et al. (1992) and Rheinfrank & Evenson (1996). This work is not discussed here but, there is a clear connection between what they refer to as a design language and what the HCI community is now referring to as a pattern language.

Tidwell's (2006) pattern collection contains patterns for desktop and web-based applications. It is split into the sections seen in Table 5. Each of Tidwell's sections contains a number of patterns which address the common design problems within these areas.

Table 5. Areas in Tidwell's Patterns

Information architecture and application structure	Navigation, signposts, and wayfinding
Layout of page elements	Actions and commands
Trees, Tables, and other information graphics	Forms and controls
Builders and editors	Visual style and aesthetics

van Welie's pattern collection is first split up into Web design patterns, GUI design patterns, and MobileUI design patterns. Table 6 contains van Welie's web design patterns.

Table 6. van Welie's Web Design Patterns

Different types of sites	Navigation
Managing Collections	Visual Design
User Experiences	Searching
Page Elements	Ecommerce
Basic Page Types	Basic Interactions

Van Duyne, Landay, and Hong's (2003) pattern language contains patterns for the design of websites. The patterns in Van Duyne et al.'s pattern language fall into one of the following groups, seen in Table 7.

Table 7. van Duyne et al.'s Patterns

Site Genres	Creating a Navigation Framework
Creating a Powerful Homepage	Writing and Managing Content
Building Trust and Credibility	Basic E-commerce
Advanced E-Commerce	Helping Customers Complete Tasks
Designing Effective Page Layouts	Making site search fast and relevant
Making Navigation Easy	Speeding Up Your Site

The Yahoo! Design Pattern Library is organized around user and application needs. The Yahoo! Design Pattern Library is broken up into the sections seen in Table 8.

Table 8. Yahoo! Design Patterns

User Needs Design Patterns			Application Needs Design Patterns	
Navigate	Explore Data	Perform Action	Call attention	Improve readability
Organize Data	Give Feedback	Customize	Group related items	Organize screen/page

Wellhausen (2006) describes what he refers to as a pattern language for user interface design for searching. The pattern language consists of 14 patterns (table 9). Wellhausen defines "Intuitive Search Interface" as a starting point for his language.

Table 9. Wellhausen's Pattern Language

Intuitive Search Interface	Refine Results
Simple and Expert Search Dialogue	Save Searches
Dynamic Search Dialogue	Search Over All Fields
Hide Data Structure	Result Count Feedback
Make Data Structure Explicit	Deactivate Fields
Freestyle Search	Static Search Dialogue
Context Searches	Search Bar

Henninger & Corrêa (2007) point to more than 400 published patterns within HCI. In addition to the many pattern collections which exist in print (Borchers, 2001; Tidwell, 2006; van Duyne et al., 2003) and on the web (Computer Professionals for Social Responsibility (CPSR); Coram & Lee; Griffiths; Hillside.net; van Welie; WebPatterns.org) there have been a number of workshops and panels devoted to patterns in HCI (Fincher, 2003b; Griffiths et al., 2000; Schummer et al., 2004; van Welie et al., 2002). In the following section we describe the various ways in which it has been suggested these patterns and pattern languages be used and the goals of using them.

2.11 Suggested Uses for Patterns and Pattern Languages

We have discussed many of the pattern collections and pattern languages which exist within HCI. Here, we discuss how these pattern languages are being used in HCI and the goals of using them. There are few studies describing how patterns have actually been used in experimental or real world settings. There have been a few empirical studies which examine the impact of patterns on design in HCI (including Borchers, 2002; Chung et al., 2004;

Dearden, Finlay, Allgar, & McManus, 2002a; Dearden, Finlay, Allgar, & McManus, 2002b; Finlay et al., 2002; and Saponas et al., 2006) or other disciplines (Golden, John, & Bass, 2005a; Golden, John, & Bass, 2005b). And there have also been a few accounts of patterns being used in real settings (Guy, 2005; Leacock, Malone, & Wheeler C., 2005). While we do not know much about how patterns are being used or how they should be used there are a number of speculations about how they might be used and why they might be used. Borchers et al. (2001) describe

> The goals of HCI design pattern language are to share successful HCI design solutions among HCI professionals, and to provide a common language for HCI design to anyone involved in the design, development, evaluation, or use of interactive systems (p. 380).

Erickson (2000) sees three characteristics which make pattern languages especially useful for design: the concrete nature of pattern languages, the way in which they are intended to be generatively used, and the interaction of the above. While Erickson points out how patterns may be useful in design, Borchers (2001) suggests that patterns may be useful in the design, evaluation, and use of systems, and in the communication about them.

Patterns as Both a Design and Evaluation Tool

Although patterns have been described mainly as a design tool they do not necessarily have to be used only to design interactive systems. Wesson & Cowley (2003) point out that "design patterns have a variety of uses, and can be used in a generative as well as an evaluative capacity" (p.1). They note that pattern based design has not been widely documented or researched but they suggest that patterns can be used to develop prototypes and to evaluate existing designs. Mahemoff & Johnston (1998a) view pattern languages as a way to bring design and evaluation closer to one another. Mahemoff & Johnston state “the best way to ensure usability is to treat human factors as an input to design” (p. 25). They also identify the need to evaluate designs instead of implementations because problems are usually easier to fix if they are discovered earlier in the design process.

Wesson and Cowley (2003) provide a comparison of patterns and guidelines as evaluation tools. Their evaluation of an e-commerce website (www.Kalahari.net) using patterns and guidelines exposed possible advantages of using guidelines in comparison with patterns. van Welie & Klaasse (2004) evaluated websites of three large museums using patterns. One of the main purposes of their study was to examine the extent to which an existing pattern collection could be used to analyze websites, specifically museum websites. Cowley & Wesson (2005) compare the use of guidelines and patterns for the design, evaluation, and re-design of websites. They report subjective ratings from 33 students to questions about using guidelines and patterns. Cowely and Wesson conclude that designers consider patterns to be an efficient and effective aid for design, evaluation, and re-design.

Although patterns have been suggested as both design and evaluation tools, we still need more empirical work in this area. While there are many possible benefits of using patterns in HCI, there are just as many problems with patterns in HCI, as they exist today. In the following section we summarize some of these problems.

2.12 Problems with Patterns in HCI

There are a number of problems identified in the literature regarding the current state of patterns and pattern languages in HCI. Some recognized problems currently being addressed by the research community are: the lack of standard format (Seffah & Javahery, 2002), the lack of an organizing principle (Fincher, 1999b; Fincher & Windsor, 2000; van Welie & van der Veer, 2003), and the lack of tools to access pattern collections (Seffah & Javahery, 2002). Another problem is the misunderstanding or misinterpretation about the difference between a pattern and a pattern language and who the intended users of a pattern language should be. While Seffah & Javahery (2002) and van Welie & van der Veer (2003) point to the text-based format of patterns and the lack of tool support for documenting and sharing patterns as compromises to the usability and accessibility of pattern languages, at this point, it is not clear if it is the lack of tool support, the lack of an organizing principle, or something else altogether that is the major compromise to usability. We clearly need more empirical work in these areas.

Seffah & Javahery (2002) suggest that the lack of a standard format for documenting patterns also hinder usability and accessibility. Fincher (2003a) points out that much of the effort surrounding patterns and pattern languages, in HCI, has focused on a search for 'the' form for individual patterns. The form a pattern should take is a valid concern which should be addressed within HCI. If we, as a community, would all like to benefit from the published patterns it would be helpful to agree on a standard format. Yet, we have no empirical support that suggests one form is better than another.

Fincher (2002) points to the problems with patterns in HCI including that there is no “language” which individual patterns might fit into and that there is no organizing principle. Fincher also acknowledges that there is no value system against which patterns are measured. Although we agree that there is no language or universal principle which all patterns may fit into, we are not sure that this would be useful or possible. Again we need more empirical work in this area.

In addition to all the problems discussed in the literature including: access to patterns, tool support, organization of patterns, and relationships between patterns, there is also the issue of pattern languages and patterns being rather general. By general we mean that the pattern collections and languages (not including van Duyne et al., 2003; Wellhausen, 2006 and a few others) do not tend to focus on any particular domain. Again, at this point it is not clear whether or not pattern languages need to be domain specific in order to be useful. All these issues point to one big problem within HCI, there is a lack of empirical work to base any of these claims on.

2.13 More Empirical Work is Necessary

In addition to all the issues discussed above, which have been recognized by the HCI community, there is one other issue which has been identified by the HCI community but not fully addressed. Currently, there is a lack of empirical evidence which suggests that patterns or pattern languages help designers design systems, or that systems designed using patterns or pattern languages are of higher quality than those designed using other means. All the other problems and challenges faced by the HCI community regarding patterns seem secondary when considering the lack of empirical studies.

We, as a community, need to focus on identifying the value in using patterns and pattern languages in HCI. At this point, there is no compelling empirical evidence that suggests that patterns or pattern languages help designers in any way throughout the design process or that they help produce high quality designs or designs that are somehow better quality than designs which have been created by some other means. All of these problems and issues which are being recognized and addressed by those in the HCI community assume that patterns and pattern languages help designers in some way or help produce quality designs. Yet, we have very little, if any, empirical support that suggests that patterns and pattern languages in any way contribute to improving the quality of the design process or the design artifact. Dearden & Finlay (2006) in their review point out

> Significant contributions have been made in the development of patterns and pattern languages which have been employed in the design of real systems…however, although the use of patterns is reported, there is little concrete evaluation of either the usefulness of pattern languages within the process or the contribution which they

> have made to the quality of the end product or to the design process (with notable exceptions) (pp. 85-86).

Carroll (2006) suggests that "patterns need and deserve a lot more work" (p. 2). Dearden & Finlay (2006) suggest a research agenda which includes “evaluating the contribution that pattern languages can make” (p. 86). Others, such as Erickson (2000) and Wesson & Cowley (2003) point out that empirical evidence is limited. In the next section we summarize, in some detail, the empirical studies which have been done to date.

2.14 Summary of Empirical Studies

There is a lack of empirical evidence which supports the notion that patterns and pattern languages are of some value in HCI. There are also very few examples of pattern languages being used in architecture or other domains. Most of the claims made about pattern languages being used seem to be based on the fact that *A Pattern Language* is a best selling book. This lack of empirical work leads us to question why we are creating patterns and pattern languages, and using patterns and pattern languages, when we do not fully understand their value in HCI. There is still much to be explored here.

There have been a few empirical studies which examine the impact of patterns on design in HCI (including: Borchers, 2002; Chung et al., 2004; Dearden, Finlay, Allgar, & McManus, 2002a; Dearden, Finlay, Allgar, & McManus, 2002b; Finlay et al., 2002; Saponas et al., 2006) or other disciplines (Golden, John, & Bass, 2005a; Golden, John, & Bass, 2005b). At some level each of these studies has an empirical component. Golden, John, & Bass (2005a) and Golden, John, & Bass (2005b) examine the use of Usability-Supporting Architectural Patterns (USAPs) while Borchers (2002), Chung et al. (2004), Dearden, Finlay, Allgar, & McManus (2002a), Dearden, Finlay, Allgar, & McManus (2002b), Finlay et al. (2002), and Saponas et al. (2006) examine the use of interaction patterns. Although Golden, John, & Bass describe experiments in which Usability-Supporting Architectural Patterns (USAPs) are used not "interaction patterns," the USAPs are a pattern-like structure and the results are interesting, therefore the studies are summarized below.

Golden, John & Bass. 2005. The Value of a Usability-Supporting Architectural Pattern in Software Architecture Design: A Controlled Experiment

Golden, John, & Bass (2005b) report on a controlled experiment designed to examine the impact of different parts of Usability-Supporting Architectural Patterns (USAPs) on the modification of a software architecture design. Each USAP consists of an architecturally sensitive usability scenario, a list of responsibilities, and a sample solution. It is clear that USAPs, as they are described, have some similarities to interaction patterns in HCI but, there are also clear differences. Eighteen graduate computer science students participated in the experiment described by Golden, John, & Bass. The experiment measured whether the USAP or subsets of the USAP supported the needs of a usable cancellation more than others. Participants in the first condition received only a usability scenario (one paragraph). Those in the second condition received the same scenario and a list of general responsibilities that should be considered in the implementation of a cancel command (three pages). Those in the third condition received the same scenario, the list of general responsibilities, and a sample solution (eight pages in total).

The participants were given task instructions along with a version of the USAP. They were also given a component interaction diagram, a sequence diagram, and a list of component interaction steps, each of which had white space for the participants to add their modifications to support cancellation. A responsibility was counted as being considered by participants if it appeared in any of the diagrams or in any lists of responsibilities added by the participants. Golden, John, & Bass (2005b) conclude that participants who received only the cancellation scenario considered, on average, a third of the responsibilities considered by those in the third condition who were given the full USAP. Their results also indicate that the full USAP increases performance without adding additional time. Golden, John, & Bass also conclude that the USAP was helpful to people with different levels of experience. The results of this study suggest that exposure to the USAPs helped the participants create better designs, in that the designs included a higher number of cancellation responsibilities.

Golden, John & Bass. 2005. Quality vs. Quantity: Comparing Evaluation Methods in a Usability-Focused Software Architecture Modification Task

Golden, John, & Bass (2005a) further analyze the modified designs that resulted from the experiment described by Golden, John, & Bass (2005b). Here, the authors examine the quality of the resulting software architecture designs. They measured quality by counting the cancellation responsibilities considered and by having architecture experts assess quality on a 7 point likert-type agreement scale.

Eight software architecture experts evaluated the materials and the canonical solution designed by the authors. According to Golden, John, & Bass (2005a) the canonical solution was not used a definitive “answer sheet” but rather the process of creating it assured that the evaluators were aware of what a reasonable solution was. Five experts evaluated 45 solutions. Most solutions were evaluated by two to four experts, but two solutions received only one expert evaluation. ANOVA showed a significant affect of the USAP on quality. Pair-wise comparisons indicated a significant mean difference between those given the scenario alone and those given the full USAP. Time on the task did not seem to have a significant impact on quality.

Golden, John, & Bass (2005a) then compared the experts’ judgments of quality with the count of responsibilities from the previous study. The authors report significant correlation between the number of responsibilities considered by participants and the quality of the solution (as judged by the experts). Golden, John & Bass conclude that the use of the full USAP increased the quality of the participants' solutions. In addition, the participants who used the full USAP identified and addressed, on average, three times as many cancellation responsibilities as those who received only the scenario, in the same amount of time, without having more experience. The authors are suggesting that the more cancellation responsibilities considered the greater the usability. It seems that in essence exposure to the full USAP helped the subjects identify more cancellation responsibilities, which were correlated with a better quality solution.

The authors suggest the full USAP with all three parts (scenario, list of responsibilities, and example solution) is similar to interaction patterns described in HCI and although they are similar, they are not exactly the same. Golden, John, & Bass (2005a) point out that their results are good news for pattern advocates, even though there was no statistical difference in coverage and quality, there was an added benefit with the addition of the example. The results of this study suggest that more cancelation responsibilities are correlated with better quality solutions (quality in terms of number of cancellation responsibilities considered and an expert quality rating).

Borchers. 2002. Teaching HCI Design Patterns: Experience from Two University Courses

Borchers (2002) reports on the use of patterns to teach HCI basics in two university computer science courses. One course took place at Stanford University in the United States and the other took place at the University of Ulm in Germany. The students at the University of Ulm received a 90 minute lecture on HCI patterns in addition to receiving a copy of Jennifer Tidwell's Common Ground pattern collection. The students studied the patterns to find patterns which could be used in their first prototyping exercise. Two weeks after the lecture on HCI patterns the students received an unexpected formal survey. According to Borchers students remembered on average 1.73 patterns (St. Dev. = 1.65). The students considered the patterns useful for understanding HCI design issues (on a scale of 1 to 5, 1= very useful, 5= completely useless), average = 1.96 (St. Dev. of 0.65) and rated the usefulness for the current project (on a scale of 1 to 5, 1=absolutely, 5=not at all) average = 2.23 (St. Dev. = 0.89). Borchers reports that students' average confidence in the use of patterns in the future (1=certainly yes, 5=certainly not) was = 1.94 (St. Dev. = 0.81).

The course at Stanford University used Borchers' (2001) *A Pattern Approach to Interaction Design* as the textbook, in addition to other readings. Students were asked to do a number of assignments including writing their own patterns, writing a pattern language as a group, and writing a reflective essay. Two measures of evaluation existed for this course, a course evaluation and the student patterns. Borchers (2002) reports that the student patterns, "were generally of fairly good quality" (p.3). He elaborates by saying that "most student patterns had a good structure that contained the right kinds of content in the right place" (p.3). Borchers reports that students had difficulty finding the appropriate level of detail and abstraction in their patterns. Interestingly, the results from the course evaluation questionnaire indicate that the course was rated slightly below other computer science courses.

Documenting patterns is a rather complicated and time consuming task. It took Alexander (1977) and a team of people years to document the patterns in *A Pattern Language*. It is not surprising that students with very little design experience would struggle with such an activity. From Borchers' comments on the structure and content of the patterns, it seems that despite the fact that the students had little experience they were still able to understand the underlying concept of a pattern and a pattern language. This suggests that patterns are something that may be easy to teach and may be used to communicate with both those outside of the field and those within the field with little experience.

Dearden, Finlay, Allgar, & McManus. 2002. Using Pattern Languages in Participatory Design.

Dearden, Finlay, Allgar, & McManus (2002b) describe their process for applying patterns languages to interaction design as a process in which a designer facilitator works with the user to develop a design where there is a phased introduction of patterns, concrete representations, for example paper prototyping, are used, and there is iterative development. Dearden et al. discuss two of the studies discussed in Finlay et al. (2002) and are summarized below.

Dearden, Finlay, Allgar, & McManus. 2002. Evaluating Pattern Languages in Participatory Design.

Dearden, Finlay, Allgar, & McManus (2002a) evaluate pattern languages in participatory design based on three criteria from Alexander's work: empowering users, generative design, and life-enhancing outcomes. Dearden et al. asked six users (with different experience levels) to create paper prototypes of a travel website using a pattern language that the authors had developed for airline and rail-travel websites. The subjects were exposed to the pattern language in various ways. Two participants were given an hour to read the pattern language, two were given the pattern language at the beginning of the session and two were given patterns as they session went on. All the users were told that they did not need to include the patterns in their designs. Dearden et al. conclude that a pattern language, as limited as the one they used, may empower users to participate in the design process. They suggest that the form of facilitation, the patterns, and the pattern language may be important issues in developing a generative process. Dearden et al. report that their results do not allow them to make claims about the quality of the prototypes. The results of this study suggest that pattern languages may be able to empower users.

Finlay, Allgar, Dearden, & McManus. 2002. Pattern Languages in Participatory Design.

Finlay, Allgar, Dearden, & McManus (2002) summarize the results from three studies. The first study involved the use of Borchers' (2001) blues language. Four users were given 11 blues patterns and were asked to generate blues music. The users had varying degrees of musical experience. Finlay et al. report that all the users were able to generate blues music from the pattern language but, they report that they were not able to judge the quality of the music. The second study, described by Finlay et al., is the study described in Dearden, Finlay, Allgar, & McManus (2002a). This study was summarized above therefore it will not be discussed here. The third study, described in Finlay et al., involved six user-designers (five had web design experience) who were asked to design a website to support the learning of oral presentation skills. A facilitator guided the user-designers through the design process and suggested patterns that might be useful throughout the session. The user-designers were informed that they did not need use the patterns. All six users-designers initially found the exercise difficult but, in the end they were all positive about the experience.

According to Finlay et al. (2002) some users indicated that the patterns gave them ideas, while a more experienced user-designer claimed that the pattern language helped him design more quickly. Finlay et al. report that the users suggested that they could not have completed the exercise without the facilitator, therefore suggesting that pattern languages may be more suitable for collaborative design rather than design done by individuals. The users reported that the examples and the bold solution text were most useful but, they thought the patterns were too detailed. Finlay et al. report that the users enjoyed the process. The results of this study suggest that users may be able to in some way use patterns, possibly with or without a facilitator.

Chung, Hong, Lin, Prabaker, Landay, & Liu. 2004. Development and Evaluation of Emerging Design Patterns for Ubiquitous Computing.

Chung, Hong, Lin, Prabaker, Landay, & Liu. (2004) describe a pattern language for ubiquitous computing that contains 45 pre-patterns. They refer to the patterns as pre-patterns because the pre-patterns are still emerging and not yet commonly used. Chung et al. claim to have conducted the first controlled study with designers using patterns. To the best of our knowledge, we agree that this is the first controlled study.

Contrary to the way Alexander explains evolving a language from observations in the real world, the authors developed their language by first brainstorming for possible pattern candidates, then by trying to find examples for their pattern candidates, followed by a card sort that was used to help organize the pattern candidates. Chung et al. (2004) then wrote the content for the patterns. After writing the patterns the authors asked other researchers to guess what a pattern was from its name. The pattern was then shown to the researchers and they were asked to rate the quality of the name and comment on the pattern as a whole. Based on this feedback the authors revised the patterns. The resulting Ubiquitous Computing (ubicomp) pattern language contained 45 pre-patterns which were grouped into one of four groups: ubicomp computing genres, physical-virtual spaces, developing successful privacy, and designing fluid actions. It is not clear why the authors grouped the patterns in such a way other than the fact that these groups resulted from the card sort.

Chung et al. (2004) conducted two rounds of evaluation. In the first round nine pairs of designers created a design for a location-enhanced application. Four pairs were exposed to the patterns and the other five were not. After the first round of evaluation the authors modified the patterns. In the second round six pairs of designers were exposed to the modified patterns and one pair was not. Chung et al. report that there were no statistically significant differences (in quality, completeness or creativity) in the first round between pairs who were exposed to the patterns and pairs who were not. In the second round they report that there were statistical differences in perceived usefulness and speed of accomplishing the task. Chung et al. also point out that most of the differences were between expert and novice designers. They conclude that their observations suggest that patterns helped novice designers produce ideas. In addition to helping those new to ubicomp learn about the field. They also suggest that the patterns helped the designers communicate ideas and avoid problems early in the design process. The results of this study seem to suggest that patterns may help novice designers in domains they are not very familiar with.

Saponas, Prabaker, Abowd, & Landay. 2006. The Impact of Pre-Patterns on the Design of Digital Home Applications.

Saponas, Prabaker, Abowd, & Landay (2006) base their work on what they and Chung et al. (2004) refer to as “pre-patterns.” The authors refer to these patterns as pre-patterns because they do not currently exist in the world instead they were created by the authors. Although Saponas et al. acknowledge this, it is important to note that really good solutions to some problems may only exist after years of evolution.

Saponas et al. (2006) asked pairs of professional designers to create interfaces for a home food inventory system. Half of the design teams had access to the 48 pre-patterns through a web interface. This group was familiarized with the pre-pattern’s content and browser prior to the task by viewing a five minute video and a 10 minute exploration period, they were also given a quiz that ensured a baseline comprehension. After the design task participants filled out a background questionnaire. The patterns group was given a questionnaire assessing how they felt about the pre-patterns.

Each design was then subjected to a heuristic evaluation by three external expert evaluators who also scored the designs using three subjective measures: detail, completeness and quality (Saponas et al., 2006). Saponas et al. developed a coding scheme for the four distinct activities they identified: discovery, idea Generation, issue Clarification, and re-reference. Re-reference and Discovery activities accounted for 17% of total activity while Idea Generation accounted for 31% and Issue Clarification accounted for 35%. Saponas et al. report that 90% of the 22 designers agreed that the pre-patterns would be useful for designing for unfamiliar domains, creating more complete designs, and aiding in the creation of a shared vocabulary. They report that the designers were most negative about the pre-patterns being enjoyable to use and suggest that this was mainly caused by the difficulty in understanding the pre-pattern relationship visualization.

As a result of the heuristic evaluation the evaluators found 255 issues in the control group’s designs and 222 issues in the pattern group’s designs. Their research team then combined the violations identified by more than one evaluator. After the consolidation there were 220 unique items in the control group and 183 unique issues in the patterns group. Saponas et al. (2006) point out that when considering issues for which the average severity rating was a four or five, the difference between the group’s decreases. The authors categorized the issues into specific heuristics violated and found that the patterns group had fewer “Attention” and “Conceptual Model” heuristic violations than the control group. Saponas et al. report that the scores for the control group and the patterns group were not statistically significant for completeness and overall perceived quality but the control groups detail scores were higher than the patterns group.

The heuristic evaluation yielded a higher mean score for teams in the control group but the two conditions showed no statistical difference in terms of high severity issues. The fact that there were less conceptual model violations in the pre-patterns group led the authors to believe that the pattern group designs were more fundamentally sound. Saponas et al. (2006) argue that the results suggest that pre-patterns allow designers to eliminate more issues early in the design process. They conclude that their study suggests that pre-patterns have a positive effect on early stage design.

The results of this study suggest that, based on the participants subjective ratings, designers believe pre-patterns will be helpful for designing in unfamiliar domains, in creating more complete designs, and creating a shared vocabulary. Saponas et al. (2006) coding of the designers’ communication suggests that patterns may help designers communicate while the comparison of the quality ratings does not strongly suggest that patterns helped improve the quality of the designs in any way.

2.15 Summary and Rationale for Research Questions

The empirical studies published to date have each made a contribution to our understanding about the value of patterns and pattern languages in HCI but, further work is necessary.

Rationale for Research Questions

In Dearden & Finlay’s (2006) review of pattern languages in HCI they conclude with a prioritized research agenda surrounding four areas:

- Exploring appropriate ways to use pattern languages in design and in education, and evaluating the contribution that pattern languages can make.
- Finding ways to organize pattern languages in HCI so that the patterns at different levels (from the broader social context of systems to the detailed of interfaces) can be applied together in design.

- Exploring and improving the processes by which patterns are identified, recorded and reviewed so that the existing stock of patterns and pattern languages available in HCI can be constantly approved and enlarged, in particular to include generic problems as well as those focused on particular platforms or interaction styles.
- Examining the way that values are explicated and promulgated in pattern languages and patter-led design (p. 86)

Two of the three research questions examined in this study directly address the first area suggested by Dearden & Finlay, namely evaluating the contribution that patterns can make, on the design process. In addition, in each of the empirical studies completed to date, patterns have not been compared to any other structuring technique. Therefore it is important to compare patterns as a structuring technique to some other structuring technique. The three research questions addressed in this study are:

RQ1. Does the use of a pattern language impact the quality of the designed interfaces?

RQ2. Does the use of a pattern language impact the time it takes to design interfaces?

RQ3. How do designers view the relationships between patterns?

The first two research questions directly evaluate the contribution patterns have on the quality of the interfaces and the time it takes to design the interfaces.

3. RESEARCH METHOD

This section includes a description of the research method used in this study. The research method is described in terms of research questions, instruments, experimental design, procedures, and data collection.

3.1 Introduction

As described in the previous chapter, the HCI community recognizes the potential benefits of using patterns and pattern languages to design systems. Although the potential benefits have been recognized, there are very few empirical studies which examine the impact of using patterns or pattern languages to design system interfaces. In this section the research method, used to examine the impact of an information retrieval pattern language on the design of information retrieval system interfaces, is described. This study tests the following hypotheses to evaluate whether the exposure to an information retrieval pattern language assists the participants in designing information retrieval interfaces. The hypotheses are listed below their respective research questions.

3.2 Research Questions

RQ1. Does the use of a pattern language impact the quality of the designed interfaces?

H1. A pattern language impacts the quality of the designed interfaces

RQ2. Does the use of a pattern language impact the time it takes to design interfaces?

H2. A pattern language impacts the time it takes to design interfaces

RQ3. How do designers view the relationships between patterns?

H3. Designers sort the patterns into piles that reinforce the underlying organization of the pattern language

The independent variable in this experiment was the exposure to a structuring technique with three levels: exposure to an information retrieval system pattern language, exposure to guidelines (for user interface design), and no structuring technique. The dependent variables in the experiment were time (to complete only the design sketches) and the quality of the designed interfaces in terms of ease of use, completeness, level of detail, and overall quality.

3.3 Instruments

The following instruments were used in this study.

- Pre-task Questionnaire
- Design task
- Pattern Overview/Tutorial
- An Information Retrieval (IR) system pattern language
- Instructions for Sorting Exercise
- Guidelines Overview/Tutorial
- Guidelines
- Interface Template
- Post-task Questionnaire
- Evaluation sheet

Pre-task Questionnaire

The purpose of the pre-task questionnaire was to gather both demographic information and information about the number of HCI courses the participants have completed. The pre-task Questionnaire can be found in Appendix B.

Design task

The purpose of the design task was to inform the participants of the task they were asked to complete. The design task set the stage for the study and described to the participant that they have been hired to design an information retrieval system. The design task included a brief scenario that described how a student interacts with an information retrieval system. The design task also informed the participant that the goal of the task was to design and sketch interfaces for an information retrieval system. The design task described the basic functionality that needed to be included in the information retrieval system interfaces. The design task can be found in Appendix C.

Pattern Overview/Tutorial

The purpose of the pattern overview was to be sure that the participants understood what a pattern was and how they could be used. The pattern overview can be found in Appendix D.

An Information Retrieval System Pattern Language (aIRPLane)

aIRPLane was used by the participants in the patterns condition. The IR system pattern language is described in more detail below. The pattern language can be found in Appendix E.

Instructions for Pattern Sorting Exercise

The participants in the patterns condition were asked to sort the patterns according to common card sorting practices. The instructions for this exercise can be found in Appendix F.

Guidelines Overview/Tutorial

The purpose of the guidelines overview was be sure that the participants understood what guidelines were and how it they could be used. The guidelines overview can be found in Appendix G.

Guidelines

Guidelines were used by the participants in the guidelines condition. The guidelines used were Nielsen and Molich’s (Molich & Nielsen, 1990; Nielsen, 1994a;

http://www.useit.com/papers/heuristic/heuristic_list.html) guidelines. Over the years these guidelines have slightly changed. The most simple and complete version seemed to be the version published on Nielsen's website (http://www.useit.com/papers/heuristic/heuristic_list.html). This version was given to the participants. The guidelines are described in more detail in the procedure section below.

Interface Template

The purpose of the interface template was to provide participants with a space to sketch their interfaces. The interface template can be found in Appendix H.

Post-task Questionnaire

The purpose of the post-task questionnaire was to gather participants' feelings about the design activity as a whole. The questionnaire can be found in Appendix I.

Evaluation Sheet

The purpose of the evaluation sheet was to collect the evaluators' judgment of the quality of the participants' interfaces. The evaluation sheet included definitions of each quality element in an effort to be sure all evaluators had the same understanding of what these terms meant in this study. The evaluation sheet can be found in Appendix J.

3.5 A Priori Power Analysis

An a priori power analysis was conducted to guide the number of participants included in this study. A power of .80, effect size of .20, and alpha of .05 were used in the calculation (Cohen, 1988; Murphy & Myors, 2004). The results suggest including approximately 40 or 50 subjects to achieve a power of approximately .80.

3.6 Experimental Design

An experimental between groups design was used to test the hypotheses. There was one independent variable, design technique or structuring technique, with three levels:

patterns, guidelines, and no technique. Individual participants were randomly assigned to one of the three groups. There were five dependent measures: ease of use, completeness, level of detail, overall quality, and design time.

3.7 Procedure

Individual participants completed the exercises on the Drexel University Campus. The overall procedure can be seen in Figure 6. Participants were asked to fill out a pre-test questionnaire. They were then given the design task. Those in the patterns group were individually given an overview/tutorial on patterns. They were then given the 39 patterns in aIRPLane. Those in the guidelines group were also individually given an overview/tutorial on guidelines. They were then given the 10 guidelines. Those in the control group did not receive anything other than the design task. After completing the design task participants were asked to fill out a post-test questionnaire. All the resulting designs were then evaluated by two experts. Both experts were Professors of HCI with a significant number of referred publications and decades of experience conducting related research, teaching, and designing and evaluating interactive systems.

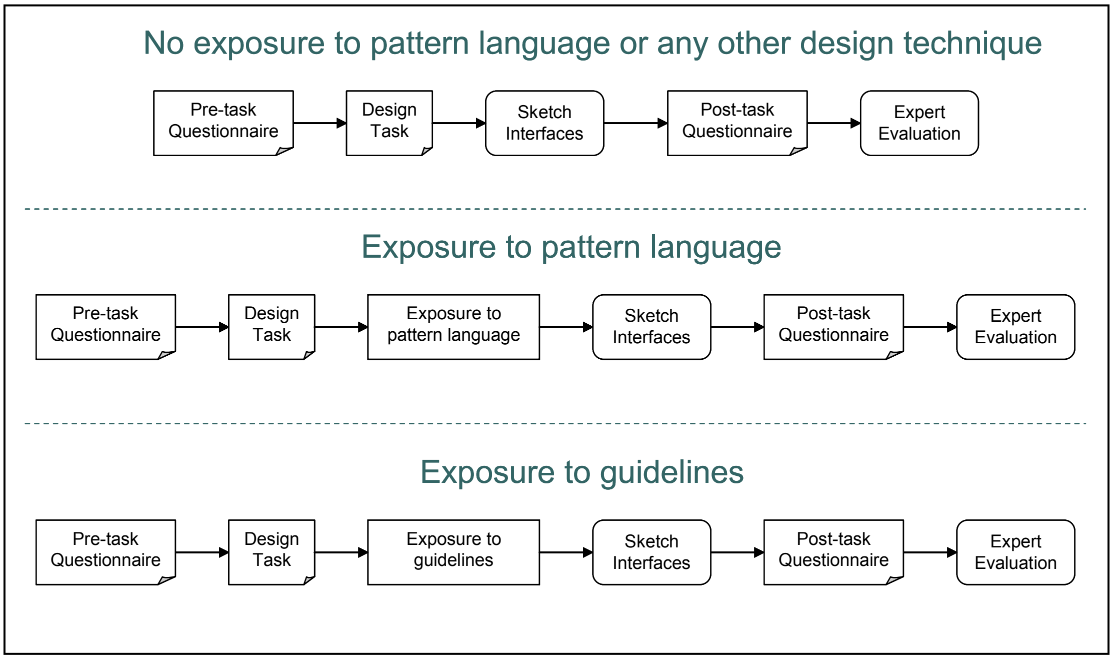


Figure 6. Experimental Procedure

Pre-test Questionnaire

Participants were asked to fill out a pre-test questionnaire (Appendix B) that gathered demographic information and information about the participants' educational experience including the number of HCI courses completed.

Design Task

Participants were given a design task (Appendix C) and informed that they were hired to design interfaces for an information retrieval system. The design task described the basic functionality that needed to be included in the system interfaces. The design task also included a short scenario which described how a student may interact with a system. The scenario was provided in order to guide the students towards sketching the required interfaces in a particular order. This was done in an effort to make the evaluation of the interfaces easier. The scenario followed a logical flow of information seeking behavior that does not seem unreasonable.

A few well known and highly cited models of information seeking behaviors were reviewed in order to reinforce the logical flow of the steps in the scenario, including Marchionini's (1995) information seeking in electronic environments, Dervin's (1983; Dervin & Nilan, 1986) sense making model, Belkin's (1980) model of Anomalous States of Knowledge (ASK), and Kuhlthau's (1991) model which addresses the affective states of information seekers. These models reinforce that the logical flow of steps in the scenario are not unreasonable.

Control Condition

The participants in the control condition were given the design task and asked to sketch the interfaces described in the design task and scenario on the interface template.

Guidelines Condition

The participants in the guidelines condition were given the design task and a brief background on designing with user interface guidelines. They were then given Nielsen and Molich's (Molich & Nielsen, 1990; Nielsen, 1994a; http://www.useit.com/papers/heuristic/heuristic_list.html) guidelines in the last format cited here. The participants were then asked if they had any questions about guidelines. If there were any questions they were addressed. The participants in the guidelines condition then sketched the interfaces described in the design task and scenario on the interface template.

Guidelines were chosen as the other design technique because they have been compared to and distinguished from patterns in the HCI literature (Granlund et al., 2001), (Henninger & Ashokkumar, 2005). (See previous Chapter for a more detailed discussion.)

Stewart & Travis (2003) point out that Nielsen's (1994a), Shneiderman & Plaisant's (2005), and Smith and Mosier's guidelines are the three most influential sets of guidelines. Nielsen and Molich's (http://www.useit.com/papers/heuristic/heuristic_list.html) guidelines have been selected as the set of guidelines to be given to those in the guidelines condition because of the reasons mentioned above and because Molich & Nielsen's (1990) claim that almost all usability problems fit into one of the categories addressed by their guidelines. Recently Nielsen (2005) addressed the claims that usability guidelines from years past may be obsolete and concluded that the majority of the guidelines from 20 years ago are still valid.

Patterns Condition

The participants in the patterns condition were given the design task and the 39 patterns in aIRPLane. They were given a brief overview on the use of patterns in HCI. The participants were then given instructions and asked to sort the patterns into piles of related patterns. They were told that there is no one right way to sort the patterns, but that there are multiple ways the patterns could be sorted because there are multiple ways the patterns are related. See Appendix F for the instructions for the sorting exercise. The participants were also asked to label or name the resulting piles of patterns using post-it notes. After completing the sorting exercise the participants were then asked to sketch the interfaces described in the design task and scenario on the interface template.

Post- task Questionnaire

After sketching the interfaces all the participants were given a questionnaire to complete. The questionnaire can be found in Appendix I. The questionnaire gathered information about the users' perceived impressions.

Evaluation of Designed Interfaces

The quality elements upon which the evaluators judged the interfaces were chosen based on definitions of usability (Nielsen, 2003) and measurements used in the empirical studies of patterns in HCI (Chung et al., 2004; Saponas et al., 2006). There is no agreed upon definition of usability, or HCI for that matter, but most would agree that usability is one of the core concepts of HCI. Definitions of usability range from a few words to multiple perspectives within a definition. Table 10 contains various definitions of usability.

Table 10. Definitions of Usability

Author	Usability Definition
Bevan, 1995	"ease of use and acceptability of a product for a particular class of users carrying out specific tasks in a specific environment" (p. 156)
Gould & Lewis, 1985	"any system designed for people to use should be easy to learn (and remember), useful, that is, contain functions people really need in their work, and be easy and pleasant to use" (p. 300)
Mayhew, 1999	"a measurable characteristic of a product's user interface that is present to a greater or lesser degree" (p. 1)
Nielsen, 2003	"usability is a quality attribute that assesses how easy user interfaces are to use… the word usability refers to methods for improving ease-of-use during the design process"
Rosson & Carroll, 2002	"the quality of a system with respect to ease of learning, ease of use, and user satisfaction" (p. 9)

Two broad dimensions of usability identified by Mayhew (1999) include how easy it is to learn the user interface and how easy it is to use the user interface. Nielsen (2003) describes the components of usability as: learnability, efficiency, memorability, errors, satisfaction, and utility. Rosson & Carroll (2002) acknowledge that it is difficult to pinpoint what makes a system usable. Rosson & Carroll describe usability as "the quality of a system with respect to ease of learning, ease of use, and user satisfaction" (p. 9).

Ease of use is described on some level in each of these definitions of usability. For this reason, ease of use will be used as one element in evaluating the quality of the resulting designs. In addition, some of the elements used by both Saponas et al. (2006) and Chung et al. (2004) will be used in order to allow for comparison to their results. Below is a list of the quality elements used in this study and their corresponding definitions. The definitions of level and detail, completeness, and overall quality were taken from Saponas et al.

- **Ease of Use** – How easy is it to use the interface?
- **Level of Detail** – How low-level, readily implementable, and non-vague are the elements of the design?
- **Completeness** – How complete is the design? Does the design contain all the necessary parts that it will need to work?
- **Overall Quality** – Overall how good of a solution is the design?

These elements have been chosen above others because we are concerned with the systems design process as a whole. Therefore we are concerned with creating interfaces (or paper prototypes) which may be handed off to someone else that may then actually implement the designed interfaces. These definitions were expanded to include examples after the first training session with the evaluators.

Evaluator Training Sessions

During the first training session it was explained to the evaluators that the interfaces would be evaluated, on a 7-point likert-type agreement scale, using the four quality elements described above (ease of use, completeness, level of detail, and overall quality). The evaluators were given the definitions of each of these quality elements. Any questions about the definitions were addressed at this point. We then walked through an evaluation of four interfaces from a pilot study using the think aloud protocol to share our ideas about why we were assigning a particular rating to each quality element. Any questions were again addressed.

The two evaluators were then presented with three different sets of interfaces from the pilot study, which consisted mainly of four interfaces: a general search interface, an advanced search interface, a results interface, and a help interface. Each evaluator was asked to individually assign a subjective rating to each set of interfaces for each quality element on a 7-point likert-type agreement scale. After each evaluator individually completed the evaluation, the evaluators disclosed their ratings and justified their reasons for assigning a particular rating. During this portion of the session it became apparent that there were particular interface elements that had an impact on the various quality ratings. These interface elements were discussed in detail. The evaluators agreed that explicitly listing some of these examples below the definitions of the quality elements would help them in their evaluations. Therefore the definitions of the quality elements were expanded to include examples of interface elements that exemplified the quality elements.

Before the second session the evaluators were sent the expanded definitions which included more detailed definitions and examples (Appendix K). During the second session the evaluators were again presented the expanded definitions of the quality elements. The evaluators were then asked to individually evaluate four sets of interfaces from the pilot study (different from those used previously) using the expanded definitions. Each evaluator was asked to individually assign a subjective rating to each set of interfaces for each quality element on a 7-point likert-type agreement scale. After completing the individual assessment the evaluators disclosed their ratings and justified their reasons for assigning a particular rating. After this discussion the evaluators were given the 52 sets of interfaces to evaluate individually. (They were both given the interfaces in the same order.)

3.8 Data Collection

In this section we discuss how data collection took place. The questionnaire (pre-task and post-task) data was collected on paper questionnaires given to the participants. The interface designs were collected on the interface template. After the participants in the patterns condition sorted the patterns into piles and named the resulting piles this data was entered into a Microsoft Word document. For each pile the name or title of the pile was input and the patterns which were sorted into that pile were named below the title. This was done for each pile which resulted from the sorting exercise. The participants' perceptions of the exercise were collected on the questionnaire using a 5-point likert-type agreement scale. The evaluator's subjective quality judgments were collected on the evaluation sheet using a 7-point likert-type agreement scale.

4. AN INFORMATION RETRIEVAL PATTERN LANGUAGE

A pattern language for information retrieval systems (aIRPLane) has been discovered by examining 30 information retrieval systems over a period of five months (3/2006 – 8/2006). This pattern language was developed due to lack of a domain specific pattern language that would lend itself to a controlled experiment. In examining the 30 systems, as patterns and their relationships were discovered they were documented and used to construct aIRPLane. Each pattern consists of a name, a picture showing the pattern, and sections titled: what, use when, why, how, examples, and how this pattern is related to others. The information retrieval pattern language can be found in Appendix E. The list of IR systems examined can be found in Appendix L.

There are few published pattern languages which could be used to design a complete set of interfaces, for example, a pattern language for a political campaign website or an e-commerce website. Indeed these websites would require some different patterns, although some patterns, for example, navigation patterns, may exist in both types of sites. Many of the published pattern languages within HCI are more like pattern collections. Therefore, it would have been difficult to ask participants to design a complete set of interfaces from the available published patterns that do not have a clear focus on a particular domain. We recognize that there are some more complete and focused pattern languages. For example, van Duyne et al. (2003) have a pattern language for the design of sites. But allowing participants to choose the type of website to design would make it much harder to evaluate the designs. This could also possibly introduce bias from the evaluators, for example, just because the evaluator is interested in ecommerce more than politics the evaluator may rate an ecommerce website design higher than a political website.

IR was chosen as the domain to examine because of the freely available access to so many IR systems (through the Drexel University library) and access to knowledgeable faculty in this area. It is also relatively safe to assume that all the participants would be familiar with IR systems, in some respect. In addition, we had knowledge of Wellhausen's (2006) pattern language for searching and saw an area in which we could expand upon and complement what had already been done. Specifically we believed an information retrieval pattern language could by greatly improved by:

- providing more coverage of the area
- making relationships explicit
- providing patterns at different levels of scale
- providing more examples

In discovering and documenting the information retrieval pattern language described herein we focused on these areas. In addition, we discovered this pattern language from existing systems whereas in the case of Wellhausen's pattern language it is not clear whether the pattern language was discovered or simply derived from experience.

4.1 The Discovering and Construction of aIRPLane

To the best of our knowledge, in the HCI literature, there are no formalized procedures for the discovery and documentation of a pattern language. As described in Chapter 2, many of the authors of pattern languages and collections in HCI describe that they have encountered the patterns throughout their years of experience. We did not want to rely on years of experience therefore we tried to develop a process that reflected the process described by Alexander (1979).

Alexander (1979) points out that patterns may be discovered in different ways, by identifying a problem and later finding a solution or by seeing a positive set of examples and therefore recognizing a solution. Alexander (1979) describes how this is a process of discovery, “a pattern is a discovery in the sense that it is a discovery of a relationship between the context, forces, and relationships in space” (p. 259).

aIRPLane has been constructed by interacting with 30 IR systems over a period of six months. The process by which aIRPLane was constructed is seen in Figure 7.

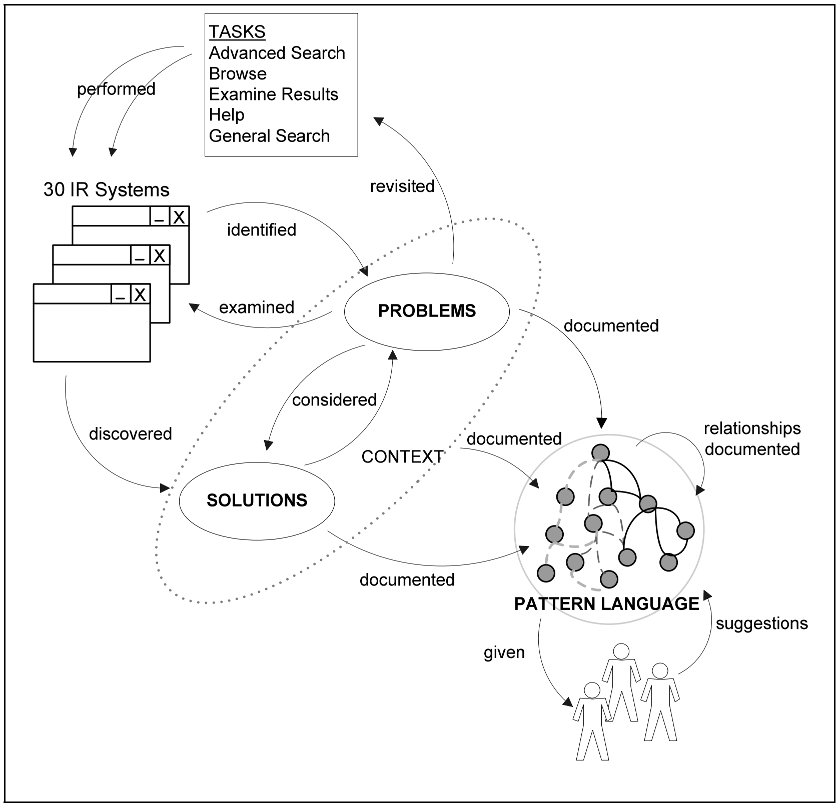


Figure 7. The Discovery and Construction Process

The following steps were used to discover and document aIRPLane:

- A list of all the IR systems available through the Drexel University library was consulted

- All the IR systems available through the Drexel University library were examined
- IR systems which were restricted and very specialized (for example, accessible to law students and faculty only) were eliminated from the list of IR systems to use
- A list of common tasks to complete using each system was created
    - The tasks were selected based on an understanding of well known information seeking models of user behavior (Belkin, 1980; Dervin & Nilan, 1986; Kuhlthau, 1991; Marchionini, 1995)
- The following tasks were performed on the 30 IR systems remaining on the list
    - Perform general search for interaction patterns
    - Examine results
    - Look at help
    - Perform advanced search for interaction patterns and human computer interaction
    - View results
    - Browse contents of system (for interaction patterns)
- As patterns were discovered a name was selected to describe each pattern
    - Patterns were usually discovered by encountering a problem and then revisiting other systems to see how the problem was addressed in those other systems
    - Patterns were also discovered after seeing a good solution multiple times
- The content for each pattern was added as patterns were identified
    - The content was added by revisiting the systems in which the pattern could be identified and/or the systems in which a usability problem was encountered

        - the problem, context and solution were then identified and described in detail
- Three examples were then found for each pattern (Lafreniere & Hedenskog, 2001)
- As more and more patterns were discovered the relationships between the patterns became apparent and the relationships were documented
    - The resulting pattern language network was documented
- The patterns were given to five doctoral students who were asked to look over the patterns
    - The patterns were revised based on feedback from the volunteers
        - Revisions included changes to pattern names, changes to pattern content, and changes to pattern examples
- Participants in the pilot study also provided feedback and suggestions about the pattern content

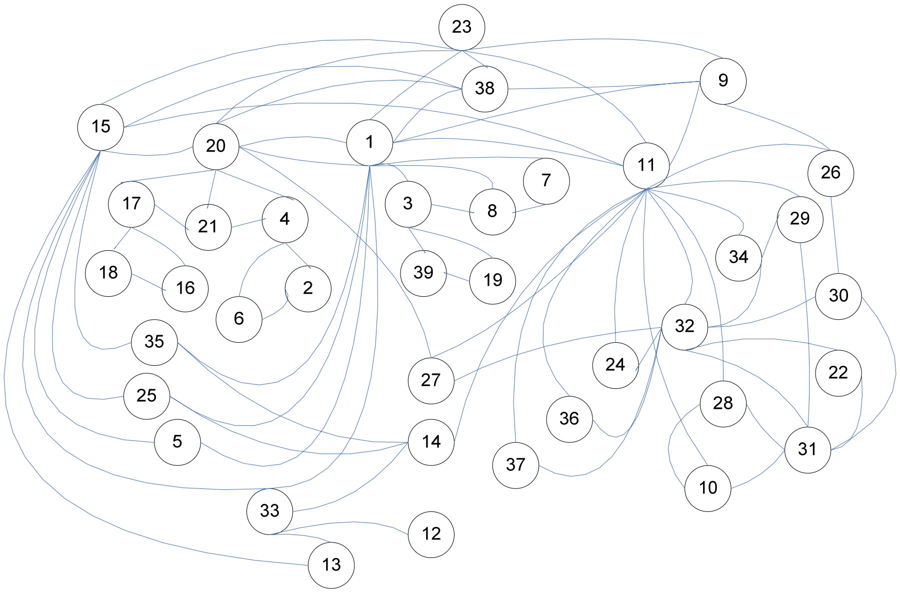
4.2 Validating aIRPLane is a Pattern Language

In Chapter 2 we described a number of ways in which others have suggested validating patterns and pattern languages. Here we compare aIRPLane to some of the various suggestions to illustrate this is a pattern language. We also note that there is no agreement on how this is to be done within the HCI community.

First, we would like to point out that aIRPLane meets the definition of a pattern language as described by Mahemoff & Johnston (2001)

> A *pattern language* is formed when a collection of patterns is arranged into a network of interdependent patterns, especially where higher-level patterns yield contexts which are resolved by more detailed patterns (p. 351).

aIRPLane is arranged into a network of interdependent patterns, as seen in Figure 8. Please note: not all possible relationships were noted in this network diagram for the sake of readability. In addition, the higher level patterns in aIRPLane, such as *general search*, are made up of lower level patterns, such as *search box and button*, *advanced search link*, *help*, and so on. It is also important to note that the place at which the lines enter the patterns in the network diagram signifies the type of relationship between the patterns. If the line enters a pattern from the top, this pattern helps make up the higher level pattern. Whereas if the line enters a pattern from the side, the patterns are related to one another is some other way.



1 Advanced_Search
2 Advanced_search_example
3 Advanced_Search_fields
4 Advanced_search_help
5 Advanced_search_link
6 Adv_search_tips
7 Adv_search_with_field_codes
8 Adv_Search_without_field_codes
9 Browse

10 Display_format
11 Display_results_
12 Execute_search_by_clicking_Go
13 Execute_search_on_enter
14 Executed_query_displayed
15 General_search
16 General_search_example
17 General_search_help
18 General_search_tips
19 Grouping_Adv_Search_fields

20 Help
21 Help_window
22 Indicate_search_terms_in_results
23 Information_retrieval_system
24 Mark_results
25 Modify_search
26 Navigation_of_results_pages
27 No_results
28 Number_of_results_per_page
29 Number_of_retrieved_results

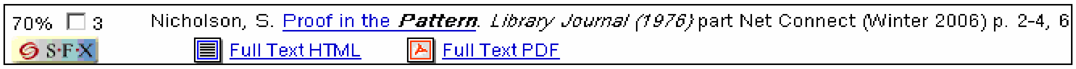
30 Position_in_results
31 Result_record
32 Result_set
33 Search_box_and_button
34 Search_history
35 Search_within_results
36 Sort_results_by
37 Sorting_order
38 Tab_Navigation
39 Tab_through_fields

Figure 8. Network Diagram of aIRPLane

Second, we would like to point out that this was a collaborative effort, as previously described. Third, we would like to compare aIRPLane with the three properties of a language, as discussed in Chapter 2. A language must have an agreed upon **set of symbols**. The set of symbols in aIRPLane consist of the 39 patterns included in the language. The meaning of the symbols is **arbitrary** and there are no clear relationships between the symbols. The symbols are arbitrary in that we could have named the patterns anything and we could change the names to anything else if we so chose. aIRPLane also contains a grammar or set of rules to manipulate the patterns. A language must contain a **grammar**, or a set of rules to manipulate the symbols and these rules can be used to create an infinite number of utterances. This can be seen in the connections designated between patterns.

In addition to meeting many of the definitions of a pattern language, we would also like to compare the patterns in aIRPLane to the definitions of a pattern. All the definitions of a pattern, reviewed in Chapter 2, include a reference to a problem and a solution and most, but not all, definitions include a reference to the context. This is what we refer to as the essence of a pattern. It is clear that the patterns in aIRPLane contain the essence of a pattern, see Figure 9.

Indicate Search Terms in Result Set



70% □ 3 Nicholson, S. Proof in the ***Pattern***. *Library Journal (1976)* part Net Connect (Winter 2006) p. 2-4, 6

Full Text HTML Full Text PDF

From http://vnweb.hwwilsonweb.com/hww/login.jhtml

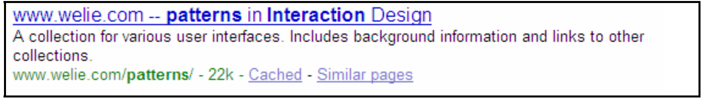
What: The user’s search terms should be differentiated from all other words in the result set.

Use When: Whenever a user’s search term appears within the result set it should be differentiated from the rest of the words in the result set no matter where it appears.

Why: The user’s attention should be drawn to the terms they included in their search query. Users are presented with a lot of information after they execute a query. The user is trying to find information related to their search query therefore when the terms a user included in their search query appear in the result set they should be indicated or differentiated in some way from all other terms in order to help users find information relevant to their query.

How: Search terms should be indicated in the result set by differentiating them from other words in the results set. This can be done by highlighting the search terms, bolding the search terms, italicizing the search terms, changing the color of the search terms, etc.

Example:

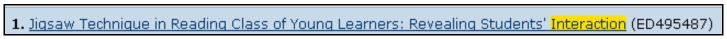


www.welie.com -- **patterns** in **Interaction** Design

A collection for various user interfaces. Includes background information and links to other collections.

www.welie.com/**patterns**/ - 22k - Cached - Similar pages

From http://google.com



1. Jigsaw Technique in Reading Class of Young Learners: Revealing Students' Interaction (ED495487)

From http://www.eric.ed.gov/ERICWebPortal/Home.portal

How this pattern is related to others: This pattern is related to *Result Set* and is a part of *Result Record*.

Figure 9. Example Pattern from aIRPLane

In addition for each of the patterns in aIRPLane, at least three instances of the pattern were found (Lafreniere & Hedenskog, 2001).

4.3 Note about aIRPLane

The information retrieval pattern language described here is just one possible information retrieval pattern language. We are in no way implying that this is the only possible pattern language for information retrieval systems. In addition, we are in no way implying that this language is complete. Pattern languages evolve over time. This work is the result of a first attempt at documenting a pattern language for information retrieval systems.

We also have a list of patterns not included in the network diagram that we would like to add to aIRPLane.

5. RESULTS

This Chapter reports the data analysis and results for each research question. This Chapter is organized in the following manner:

5.1 Introduction
5.2 Data Collection and Overview of Population
5.3 Summary of Results
5.4 Research Question 1: Impact on Quality
5.5 Research Question 2: Impact on Time
5.6 Research Question 3: The “Language” Aspect

5.1 Introduction

This study examined the impact of aIRPLane on the design of information retrieval interfaces. There were 2 primary research questions and 1 secondary question motivating this research. The first research question (RQ1) examined the impact of the exposure to aIRPLane on the quality of the information retrieval interfaces. The second research question (RQ2) examined the impact of exposure to aIRPLane on the time to design the interfaces. The third research question (RQ3) explored how the participants viewed the relationships between the patterns (by means of a sorting exercise).

Data was collected from April 2007 through August 2007. A number of instruments were used to collect data including:

- Pre-task questionnaire
- Interface design template
- Post-task questionnaire

Participants were given a hypothetical situation that described they had been hired to design an information retrieval system. Each participant used one of the two structuring techniques (patterns, guidelines) or no structuring technique (control condition) to complete the design task. Inferential statistical tests were used, where appropriate, to test the research questions.

5.2 Data Collection and Overview of Sample Population

Drexel University is located in Philadelphia, PA. Current enrollment at Drexel is 12,906 undergraduates and 6,976 graduate and professional students (academic year 2006-07). The College of Information Science and Technology, offers three undergraduate degrees and four graduate degrees. Fifty-two participants were sampled between April 2007 and August 2007. Individual participants were randomly assigned to one of the experimental conditions, Table 11.

Table 11. Experimental Design with Sample Size

Condition	Assignment	N	Treatment
Patterns Condition	Random	17	Exposure to an IR pattern language
Guidelines Condition	Random	18	Exposure to Nielsen and Molich's Guidelines (http://www.useit.com/papers/heuristic/heuristic_list.html)
Control Condition	Random	17	No exposure to any structured HCI design technique

Thirty-seven of the participants were undergraduate students at Drexel. The other 15 participants were graduate students at Drexel. The majority of the participants were Information Systems or Information Technology majors in the College of Information Science and Technology (see Table 12). Most of the participants were between 18 and 25 years of age.

Table 12. Participants College Major

	Age Range			
Major	18-25	26-35	36-45	**Total**
Information Systems	16	3	1	20
Information Technology	12	0	0	12
Information Science and Technology	6	2	1	9
Information Studies, Information Science, Library Science	1	3	0	4
Engineering (Software, Computer, Electrical, Chemical)	5	1	0	6
Digital Media	1	0	0	1
Total	41	9	2	52

On average the undergraduate students had taken 1.27 HCI courses and the graduate students had taken on average 2.06 HCI courses. Most of the participants had little if any related job experience (Table 13).

Table 13. Participants Related Job Experience

	Years of related job experience				
	0-6 months	**7-12 months**	**1-2 years**	**3 or more years**	**Total**
Undergraduate students	26	7	3	1	37
Graduate students	9	1	5	0	15
Total	35	8	8	1	52

Participants were first asked to read and sign the informed consent. They were then given the pre-task questionnaire (Appendix B). After completing the pre-task questionnaire participants were given the design task (Appendix C). After reading the design task participants were given a brief introduction to the technique they would be using, either patterns or guidelines (appendices D, G). This was first read to the participants to be sure they heard everything. Then the participants were asked to read it on their own to make sure they understood the technique and did not have any questions.

After the participants read the introduction to their technique, the participant was presented with an example of their respective technique, either a pattern or a guideline, depending on the condition they had been assigned. The examples were chosen randomly for each participant from the group of patterns or guidelines they would be given, depending on their condition assignment. The participants were asked again at this point if they had any questions. The participants were then presented with their respective technique, either the guidelines or the patterns. The participants in the patterns condition were asked to spend time reading and sorting the patterns. The same sorting instructions were read to each participant (see Appendix F). The participants in the guidelines condition were only asked to read the guidelines. They were not asked to sort the guidelines because of the small number of guidelines (only 10). On average it took approximately four minutes to introduce the participant to the design task, read the instructions, and present the appropriate technique.

The participants were not given a time limit for the sorting exercise. They were simply informed that the entire study would take approximately one hour. The average time to sort the patterns was 17.6 minutes. The average time to read the guidelines was 3.8 minutes. After completing this exercise the participants were asked to design the interfaces on the interface template (Appendix H). The average time to design the interfaces is seen in Table 14.

Table 14. Design Time

Condition	Average design time (min)	Average time on technique (min)	Average overall time (min)
Pattern	21.0	17.6	51.7
Guideline	23.4	3.8	37.4
Control	19.6	--	29.7

Once the participants completed the design they were asked to complete the post-task questionnaire (Appendix I). The questions addressed their experience in this study, as a whole, and their perception of the usefulness of the respective technique they used. Participants were then paid $15 and asked to fill out a receipt. The designed interfaces were coded with numbers so that it was not possible to determine which condition a particular participant was assigned to. A key was kept on a separate paper.

5.3 Summary of Results

Table 15 shows a summary of the results. Each of these findings is discussed in some detail in the following sections.

Table 15. Summary of Results

Research Question		Results	Statistical Test
RQ1			
	Ease of Use	No statistically significant difference	ANOVA
	Detail	No statistically significant difference	ANOVA
	Completeness	No statistically significant difference	ANOVA
	Overall Quality	No statistically significant difference	ANOVA
RQ2			
	Design Time	No statistically significant difference	ANOVA
RQ3			
	Language Aspect	Hierarchical cluster analysis dendrogram Three dimensional map R square = .77 and stress = .17	cluster analysis multi-dimensional scaling

5.4 Research Question 1: Impact on Quality

In order to answer the first research question the participants' interfaces needed to be evaluated. Examples of interfaces produced by the participants can be seen in Appendix M. Expert review was chosen as the method because of its appropriateness and use in similar studies (Chung et al., 2004; Saponas et al., 2006). After all data collection was complete we scheduled a training session with the two experts who agreed to evaluate the interfaces. Both experts were Professors of HCI with a significant number of referred publications and decades of experience conducting related research, teaching, and designing and evaluating interactive systems. After all the interfaces were rated by both experts inter-rater reliability was calculated. The experts' subjective ratings were collected on a 7-point likert-type agreement scale. Although likert-type scales are really ordinal, they are commonly considered interval, but it is important to recognize that there is not complete agreement on this issue (Jamieson, 2004; Garson, n.d.). Throughout these analyses the expert ratings on the 7-point likert-type agreement scale were considered interval.

Inter-rater Reliability

In this study SPSS was used to analyze inter-rater reliability (Appendix N). This was done using two-way mixed model, intraclass correlation (Shrout & Fleiss, 1979). Intraclass correlation was chosen because of the ability to assess consistency of ratings on interval scales. Intraclass correlation provides reliability measures when multiple judges rate the same items. The average measure was used so that the average rating, across both evaluators, could be used in further analyses. Intraclass correlation coefficient for average measures = .82 indicating a high level of agreement (Cronbach's alpha is .82).

Assessing Normality

Normality was assessed using statistical and graphical methods in SPSS. Descriptive statistics for the dependent variables including mean, median, and standard deviation are seen in Table 16.

Table 16. Descriptive Statistics for Quality Elements

	Mean	SD	Median	Min	Max
Ease of Use	3.38	1.27	3.00	1.00	6.00
Detail	3.61	1.30	3.50	1.00	6.50
Completeness	4.62	1.01	4.50	2.50	7.00
Overall Quality	3.79	1.18	3.50	1.00	6.50

Skewness and Kurtosis are two common ways to assess normality of the distribution of data. Skewness provides information on the symmetry of the distribution. Kurtosis provides information on the "peakedness" of the distribution. Skewness and kurtosis were both calculated using SPSS. Skewness values indicate the distributions are slightly skewed to the right. Kurtosis values below zero, as seen in Table 17, indicate a distribution that is slightly flat.

Table 17. Skewness and Kurtosis

	Skewness	Kurtosis
Ease of Use	.49	-.46
Detail	.32	-.55
Completeness	.25	-.31
Overall Quality	.32	-.24

Considering the Skewness and Kurtosis values suggest the distribution is slightly skewed to the left and slightly flat, we investigated the outliers and their possible influence by examining the trimmed mean, see Table 18. To obtain the 5% trimmed mean SPSS removes the top and bottom 5% of the cases and then recalculates a new mean value. By comparing the original mean and the 5% trimmed mean it is possible to determine if some of the more extreme scores are having a lot of influence on the mean (Pallant, 2004).

Table 18. Five Percent Trimmed Mean

	Mean	5% trimmed Mean
Ease of Use	3.38	3.35
Detail	3.61	3.58
Completeness	4.62	4.60
Overall Quality	3.79	3.78

The 5% trimmed means for all the quality elements are very similar to the means suggesting there are no outliers having a great influence on the mean. In addition the box plot (Appendix O) does not show any outliers. Therefore there is no further need to investigate the normality and outliers. Based on these analyses we assume the data meets the assumption of normality.

Exploring Relationships among Variables

The relationships between ease of use, completeness, level of detail, and overall quality were investigated using Pearson's correlation coefficient (as previously mentioned the 7 point likert-type agreement scale is being treated as interval). Preliminary analyses were performed to ensure no violation of the assumptions of normality, linearity, and homoscedasticity (see Appendix O). There was a very strong positive correlation between the ease of use and level of detail variables, (r= .93, n=52, p=.00) with higher ease of use ratings associated with higher detail ratings. All other correlations were also very strong, see Table 19 for all other correlations.

Multicollinearity

Multicollinearity refers to the case when the dependent variables are very highly correlated (greater than .90) as seen in Table 19. When this occurs the analyses may actually be weakened because of inflated error (Tabachnick & Fidell, 1996). Some suggest omitting one or more of the variables while others suggest creating a composite score from the redundant variables (Pallant, 2004; Tabachnick & Fidell, 1996). MANOVA is an appropriate technique when the dependent variables are moderately correlated (greater than 0.2 and less than 0.8) (Pallant, 2004; Tabachnick & Fidell, 1996). Therefore MANOVA is not an appropriate technique here.

Table 19. Correlations of Dependent Measures

	Ease of Use	Level of Detail	Completeness	Overall Quality
Ease of Use				
Correlation Coefficient	1.00	.93	.87	.97
Sig	.00	.00	.00	.00
Level of Detail				
Correlation Coefficient	.93	1.00	.82	.95
Sig	.00	.00	.00	.00
Completeness				
Correlation Coefficient	.87	.82	1.00	.90
Sig	.00	.00	.00	.00
Overall Quality				
Correlation Coefficient	.97	.95	.90	1.00
Sig	.00	.00	.00	.00

Usually in further analyses the dependent measures would be combined into a single measure or one or more of the variables would be omitted but, for the sake of completeness here, all four quality variables are included in further discussions and analyses.

Comparing Groups

The three experimental groups' quality scores were compared using ANOVA. The analysis for each quality element is discussed separately. Each of these variables is described in detail in Chapter 3. The detailed results from SPSS for all further analyses in this Chapter can be found in Appendix P.

Ease of Use

A one-way between-groups analysis of variance was conducted to explore the impact of a design technique on quality of the designs, as measured by ease of use. The independent variable was condition with three levels, patterns, guidelines and control. The significance level for Levene's test for equal variances = .82, therefore we can assume equal variances. There was no statistically significant difference at the p < .05 level in ease of use scores for the three groups F (2, 49) = .32, p = .73. The actual difference in mean scores between the groups was quite small, see Figure 10. The observed power = .10 suggests insufficient power of the test but, as suggested by (Hoenig & Heisey, 2001; O'Keefe, 2007) the observed power will always be low when there is a non significant p value because of the relationship between observed power and p values. Therefore, in this case, reporting the observed power does not add much to the analysis.

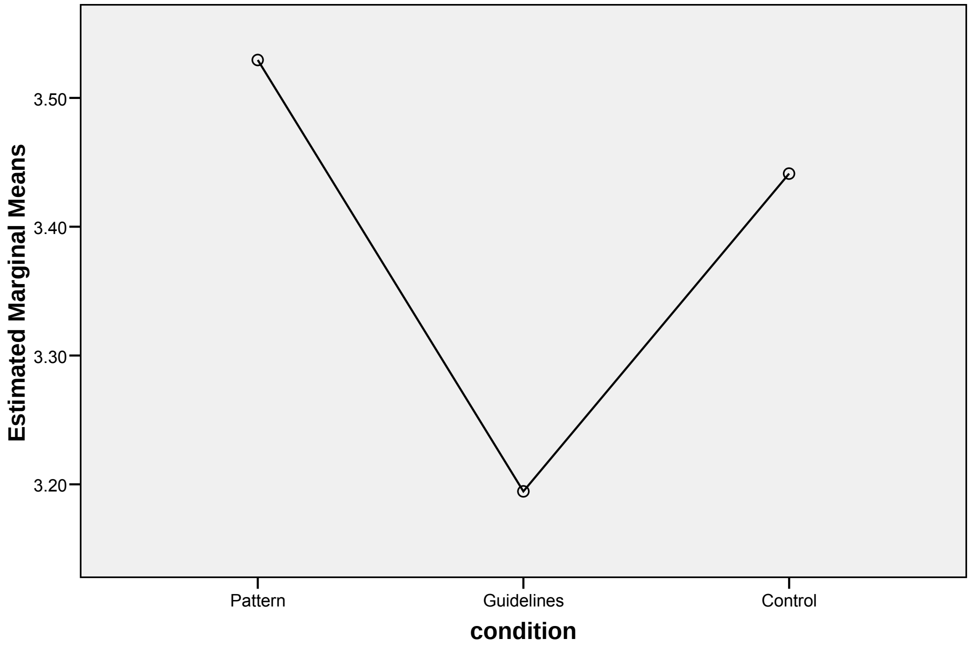


Figure 10. Means Plot Ease of Use

Level of Detail

A one-way between-groups analysis of variance was conducted to explore the impact of a design technique on quality of the designs, as measured by detail. The independent variable was condition with three levels, patterns, guidelines and control. The significance level for Levene's test for equal variances = .24, therefore we can assume equal variances. There was no statistically significant difference at the p < .05 level in ease of use scores for the three groups F (2, 49) = .35, p = .70. The actual difference in mean scores between the groups was quite small, see Figure 11.

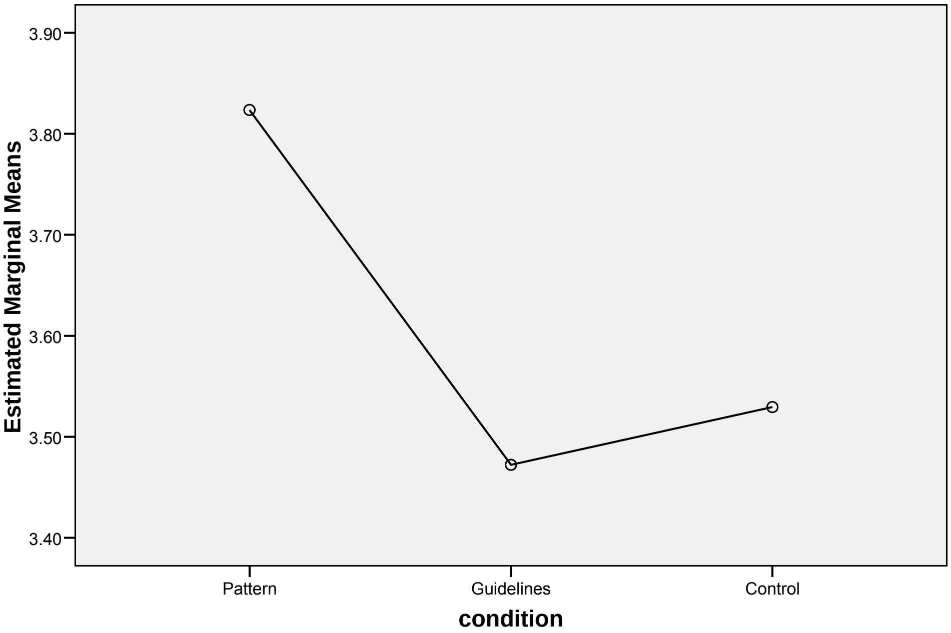


Figure 11. Means Plot Level of Detail

Completeness

A one-way between-groups analysis of variance was conducted to explore the impact of a design technique on quality of the designs, as measured by completeness. The independent variable was condition with three levels, patterns, guidelines and control. The significance level for Levene's test for equal variances = .94, therefore we can assume equal variances. There was no statistically significant difference at the p < .05 level in ease of use scores for the three groups F (2, 49) = 1.50, p = .23. The actual difference in mean scores between the groups was moderate, see Figure 12.

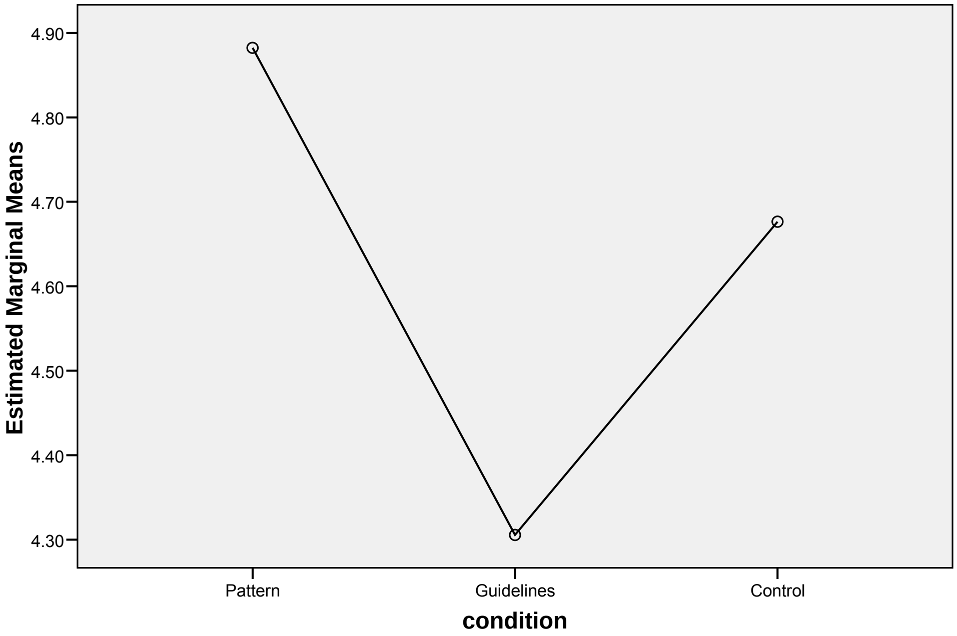


Figure 12. Means Plot Completeness

Overall Quality

A one-way between-groups analysis of variance was conducted to explore the impact of a design technique on quality of the designs, as measured by overall quality. The independent variable was condition with three levels, patterns, guidelines and control. The significance level for Levene's test for equal variances = .74, therefore we can assume equal variances. There was no statistically significant difference at the p < .05 level in ease of use scores for the three groups F (2, 49) = .63, p = .54. The actual difference in mean scores between the groups was quite small, see Figure 13.

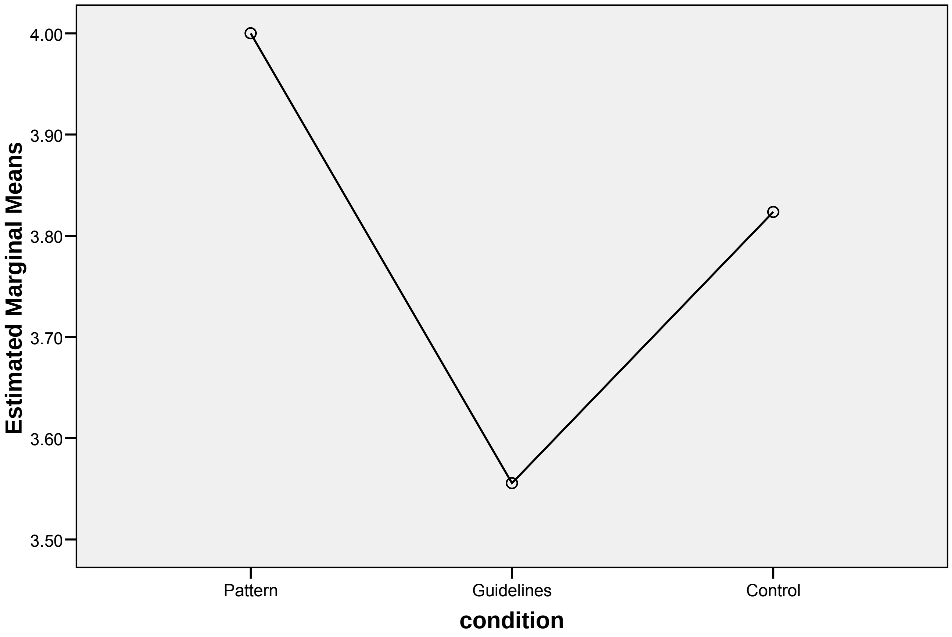


Figure 13. Means Plot Overall Quality

Summary of Comparison of Groups Quality Scores

Although there are no statistically significant differences in the quality ratings there seemed to be a trend in the data. As seen in the above comparisons the structuring techniques had little or no affect on the quality of the interfaces. The question that seems to follow naturally here is, if there is no difference between the groups, did any participants actually use the patterns in their designs, whether they were aware of it or not?

In order to further explore the role patterns play in interactive systems we revisited the participants' interfaces, the 30 systems that were examined in the discovery of an information retrieval pattern language, and the top five search engines as rated by Nielsen/Net Ratings and reported in Search Engine Watch (Burns, Oct 26, 2007) to identify the patterns that exist in those designs. This was done in an effort to better understand the extent to which the patterns in an information retrieval pattern language were used by the participants in all three conditions and to further understand the extent to which these patterns exist in real systems. As this was not intended to be part of the analysis it is not discussed in the results section, instead it is addressed in a separate chapter, Chapter 6.

5.5 Research Question 2: Impact on Time

In this section the focus is on the time it took to design the interfaces. A one-way between-groups analysis of variance was conducted to explore the impact of a design technique on the time to create the designs. Individual participants were randomly assigned to the either patterns condition, guidelines condition, or control condition. The significance level for Levene's test for equal variances = .05, therefore we have strictly violated the homogeneity of variance assumption but, because ANOVA is reasonably robust to violations of this assumption, provided the group size is similar (Pallant, 2004); therefore ANOVA is still an appropriate technique.

The average design times for the three groups are seen in Table 20. There was no statistically significant difference at the p < .05 level in design time for the three groups F (2, 49) = .63, p = .54. The actual difference in mean scores between the groups was quite small.

Table 20. Descriptive Statistics for Time

	Design Time		**Overall Time**		**Time on Technique**	
	Mean	St Dev	Mean	St Dev	Mean	St Dev
Patterns	21.00	7.46	51.71	12.35	17.65	7.22
Guidelines	23.34	13.55	37.44	12.40	3.83	7.22
Control	19.59	8.14	29.71	9.75	--	--

Relationship between Quality Ratings and Design Time

In order to further explore RQ1 and RQ2 we examined the relationship between the quality ratings and the time to design the interfaces using Pearson's correlation. There was a positive correlation between design time and each of the four quality elements, with longer design times associated with higher quality ratings, see Table 21.

Table 21. Correlation between Design Time and Quality Elements

Design Time				
	Ease of Use	**Detail**	**Completeness**	**Overall Quality**
Correlation	.40	.36	.35	.41
Sig	.01	.01	.01	.01

Summary of Comparison of Groups Design Time

As seen in the above comparisons the structuring techniques had little or no apparent impact on the time to design the interfaces. However, there was a significant positive correlation between the four quality elements and design time, with longer design times associated with higher quality scores.

Those in the patterns group spent considerably more time reading and sorting the patterns when compared to the guidelines group, but the amount of material the participants were required to read should be considered here. The participants in the patterns group looked at 39 patterns that spanned approximately 60 pages while the participants in the guidelines group read 10 guidelines that spanned nearly two pages. It is therefore not surprising that the participants in the patterns group spent considerably more time on the technique.

5.6 Participants' Subjective Perceptions

After completing the design task the participants were asked a number of questions about their experience using the respective technique and about their experience as a whole. Their responses were collected on a five-point likert-type agreement scale. The participants in both the guidelines condition (average 3.83 out of 5) and the patterns condition (average 4.35 out of 5) found the respective technique useful for the design task. Participants were also asked whether they would use their respective technique in the future. Participants in both the guidelines condition (average 4.06 out of 5) and the patterns condition (average 4.24 out of 5) said they would use the technique in the future.

5.7 Research Question 3: The "Language" Aspect

In this section the results of the sorting exercise are discussed. The purpose of the card sorting exercise was to allow us to further understand the overall structure of the language and the relationships between the patterns within the language, as seen by the participants in this study. After receiving the design task and the tutorial, but before beginning the design task, participants in the pattern condition were asked to sort and categorize the patterns using common card sorting techniques (Cooke, 1994; McCain, 1990). Card sorting has been used to elicit knowledge in a variety of domains, including psychology, education, linguistics, information science, and HCI (Hannah, 2005; Maurer & Warfel, 2004; Nielsen & Sano, 1994; Rugg & McGeorge, 1997; Tullis & Wood, 2004; Upchurch, Rugg, & Kitchenham, 2001).

The 39 patterns in aIRPLane were given to the participants in random order. The participants were instructed that there was no right way to sort the patterns. They were told that they could create as many piles as they would like with as many patterns as they would like in each pile (see Appendix F for sorting instructions).

To further understand the “language” aspect, the sort data resulting from having the participants sort the patterns was input in a co-occurrence matrix. The resulting co-occurrence matrix was converted to a matrix of correlation coefficients. This matrix of correlations was used in all further analyses including: hierarchical cluster analysis, factor analysis, and multidimensional scaling (MDS). This was done to examine the participants’ overall perception of the pattern language and the relationships between patterns. The results of the hierarchical cluster analysis as a dendrogram are seen in Figure 14. All hierarchical cluster analyses begin with a set of objects and, step by step, join objects and clusters until one cluster is achieved. The dendrogram shows the cluster structure, beginning with 39 patterns on the left and ending with a single cluster on the right. Patterns or clusters are joined based on the distance criterion from the co-occurrence matrix. The horizontal distance traveled between the merging between patterns or clusters is evidence of the integration or isolation of the patterns or clusters.

A Principal Components Analysis (PCA), with an orthogonal (varimax in SPSS) rotation of the extracted factors produces factors that are uncorrelated. The factors are interpreted based on the patterns that have high loadings, greater than +/-0.7 (McCain, 1990). The results of orthogonal rotation of the extracted factors, with a stopping rule of eigenvalue < 1, are shown in Appendix P. The results of the orthogonal (varimax in SPSS) rotation reveal 5 factors. An oblique factor rotation (oblimin in SPSS) of the extracted factors produces factors that are intercorrelated. The results of the oblique rotation with a stopping rule of eigenvalue < 1 can be seen in Appendix P. The PCA, with an oblique rotation also reveals 5 factors. Inspection of the dendrogram and the results of the PCA suggest that a 5 cluster solution is a good representation of the structure of this data set (see McCain, 1990 for discussion).

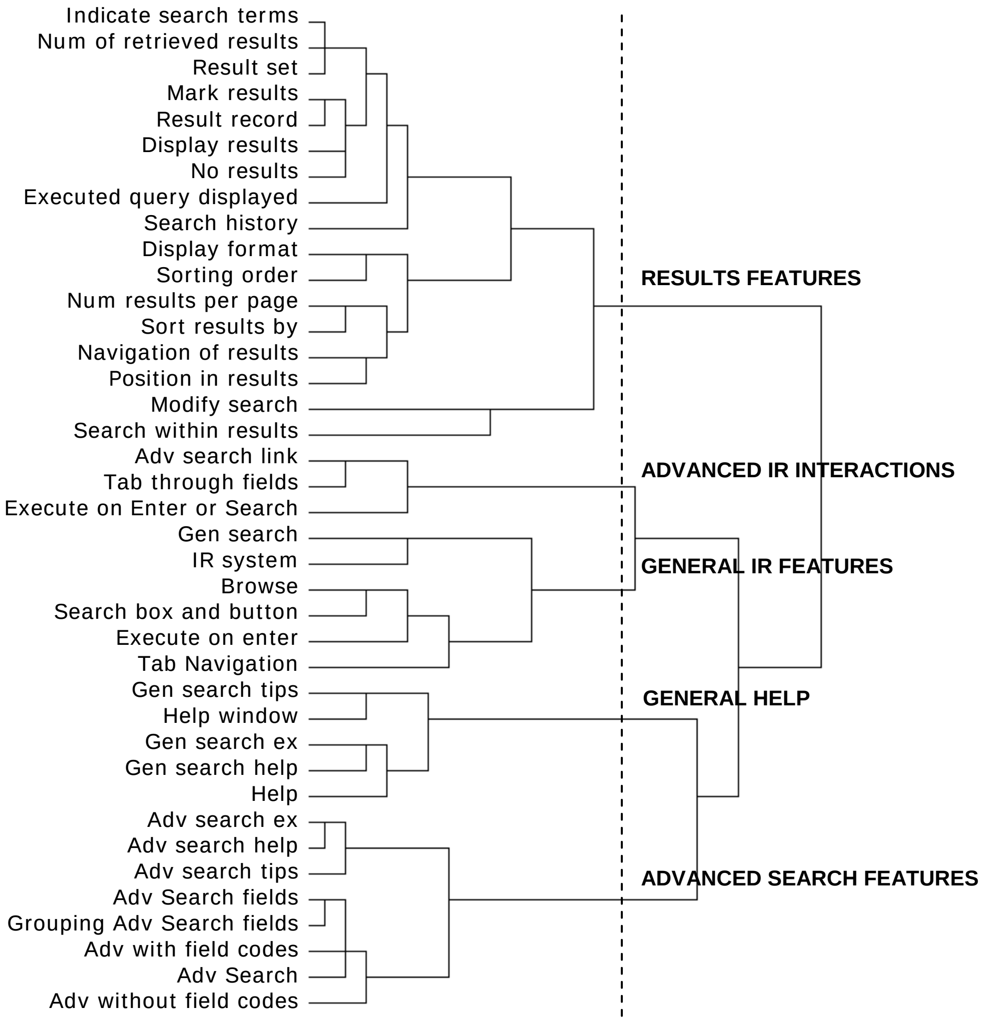


Figure 14. Cluster Analysis

The results of a two-dimensional MDS map with clusters enhanced from the hierarchical cluster analysis can be seen in Figure 15. MDS attempts to represent the whole data matrix as a two-dimensional (or more) map. In MDS, R square and stress are indicators of the overall “goodness of fit.” The R square is the proportion of variance explained. The stress is the distortion or noise in the analysis. Stress less than 0.2 is usually acceptable (McCain, 1990).

When the patterns are mapped in a two-dimensional map, as seen here, the R square = .66 and stress = .26 (Young's S-stress formula 1 is used). The two-dimensional map therefore accounts for 66% of the variance. In information science it is acceptable to force the map into more than two dimensions if the variance explained increases by more than 5% (McCain, 1990).

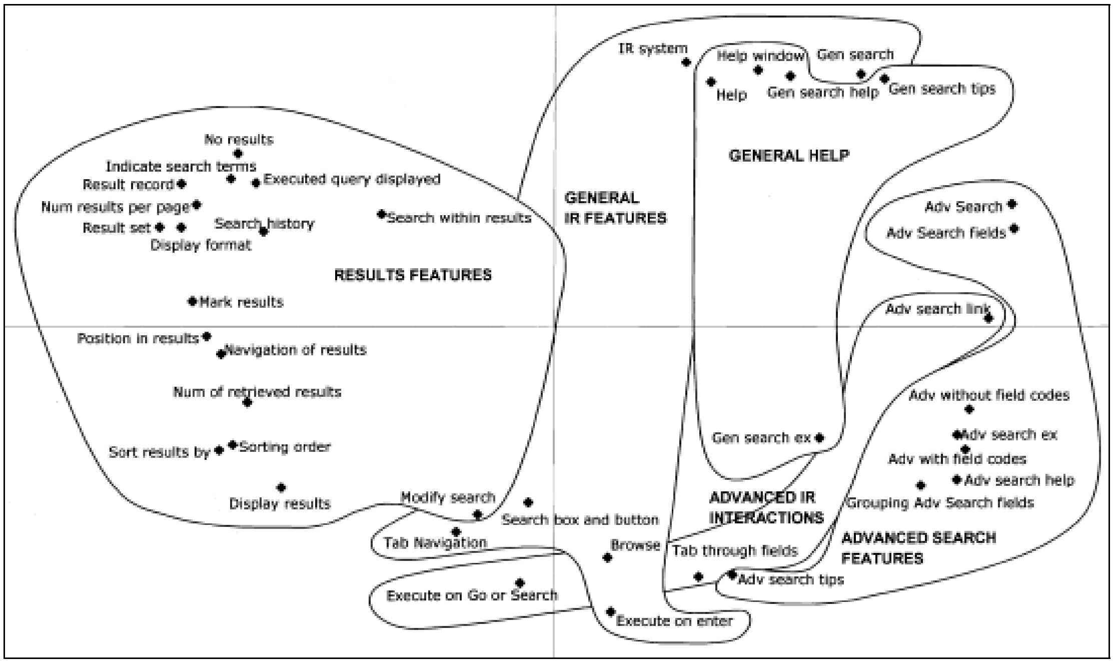


Figure 15. Two Dimensional MDS Map

In the two dimensional map seen here, patterns, represented as points on the MDS map, are positioned based on the correlation matrix derived from the pattern co-occurrence matrix. The corresponding labels on the map are the pattern names. Patterns with similar co-occurrence patterns are placed near each other in the map. Those with many links to others tend to be placed near the center of the map while highly dissimilar patterns are placed at a distance and those with few local links are at the periphery. Patterns closely positioned but placed in different clusters have important secondary links. Looking at the 2D map, in Figure 15, moving along the y axis from top to bottom we can see that the patterns go from higher level functionality type patterns, such as *help*, to lower level interaction type patterns, such as *execute on enter*. Moving along the x axis, from left to right we see the patterns go from performing the search to finding help about performing a search.

When the patterns are mapped in a three-dimensional map, as seen in Figure 16, the R square = .77 and stress = .17 (Young’s S-stress formula 1 is used). Other rotations of the 3D map can be seen in Appendix P. The three-dimensional map accounts for 77% of the variance. This suggests the participants tend to see a common arrangement of the patterns (i.e. a pattern language).

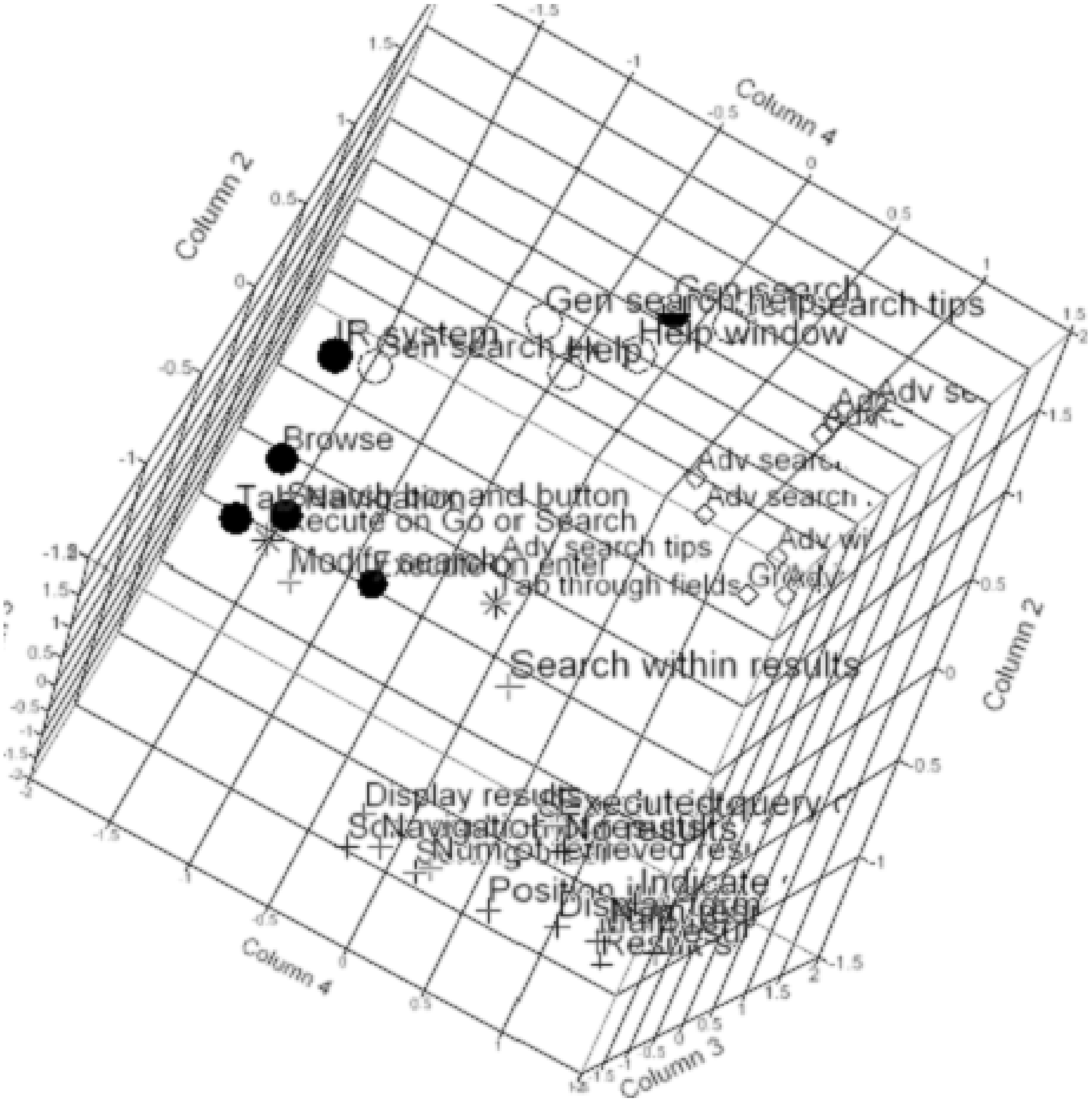


Figure 16. Three Dimensional MDS Map

The three-dimensional map, seen in Figure 16, has also been enhanced with the clusters identified earlier. Here the cluster membership is designated using different marker shapes. Those identified as being a part of the *Results Features* cluster have a “+” marker. Those indentified as part of the *General IR Features* cluster have a solid circle marker. Those in the *General Help* cluster have a solid circle marker. The patterns in the *Advanced IR*

Interactions cluster have a "*" marker. Those identified as part of the *Advanced Search Features* cluster have a hollow diamond marker. In order to compare this with the network map we have similarly identified cluster membership using the same markers, Figure 17.

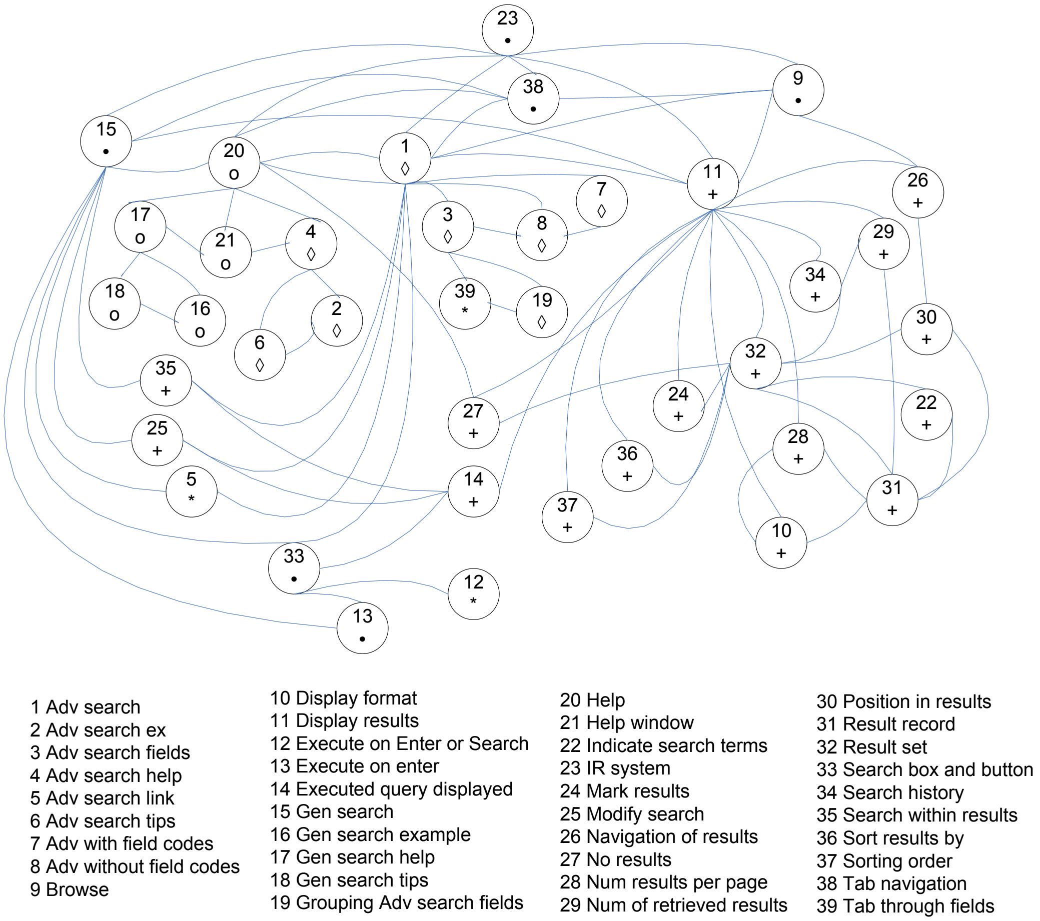


Figure 17. aIRPLane Network Map with Cluster Membership Identified

Inspection of the MDS maps and the network map presented earlier, now enhanced with the cluster membership, suggests that the participants viewed the relationships between the patterns in a similar manner to the way we viewed the relationships between the patterns prior to conducting this study. This can be clearly seen if the patterns in each cluster are identified on the network map as seen above.

Summary of Exploration of the "Language" Aspect

The purpose of the card sorting exercise was to allow us to further understand the overall structure of the language and the relationships within the language, as seen by the participants in this study. The results of the card sorting exercise, as analyzed using hierarchical cluster analysis and MDS suggest the participants tend to see a common arrangement of the patterns, in that the variance explained was 77%. This common arrangement suggested by the cluster analysis, factor analysis, and MDS also has clear similarities with arrangement of the pattern in the network map.

5.8 Summary of Results

As described above the ANOVAs used to address RQ1 did not yield any statistically significant differences between the groups for any of the 4 quality variables. The ANOVA performed to address RQ2 did not yield any statistically significant differences between the groups for the time to design the interfaces. However, there was a significant positive correlation between the four quality elements and design time, with longer design times associated with higher quality scores. The results of the card sorting exercise, as analyzed using hierarchical cluster analysis and MDS performed to address RQ3, suggest the participants tend to see a common arrangement of the patterns, in that the variance explained was 77%. The results of these analyses will be discussed more in Chapter 7.

6. FURTHER EXPLORATION OF THE RESULTS

In an effort to better understand the extent to which the patterns in aIRPLane were used by the participants in all three conditions and to further understand the extent to which these patterns exist in real systems, we revisited the participants' interfaces, the 30 systems that were examined in the discovery of an information retrieval pattern language, and the top five search engines as rated by Nielsen/Net Ratings and reported in Search Engine Watch (Burns, Oct 26, 2007) to identify the patterns that exist in those designs.

6.2 Identifying Patterns in IR systems and web search engines

The 30 systems used to discover aIRPLane were re-examined to identify which patterns exist in the systems. In order to be able identify the patterns in any system interface, the evaluator needs to be very familiar with the pattern language; because of this, we conducted the evaluation. This was done from October 31, 2007 through November 13, 2007. These dates are reported because some of the system interfaces have changed since this evaluation was done. Screens were also captured from each of these systems during the pattern identification process in case inspection was necessary at a later date. The patterns that existed in these interfaces were noted on the pattern checklist which was simply a list of the patterns in aIRPLane organized in a fashion that resembles the overall organization of aIRPLane. We systematically looked first at the overall system interface, then the general search interface. An example of a few of the patterns in aIRPLane identified in Google's general search interface can be seen in Figure 18. The patterns which existed in these parts of the systems were noted. At this point a simple query was executed and subsequently the results page was viewed. The patterns which existed in the results page were then noted.

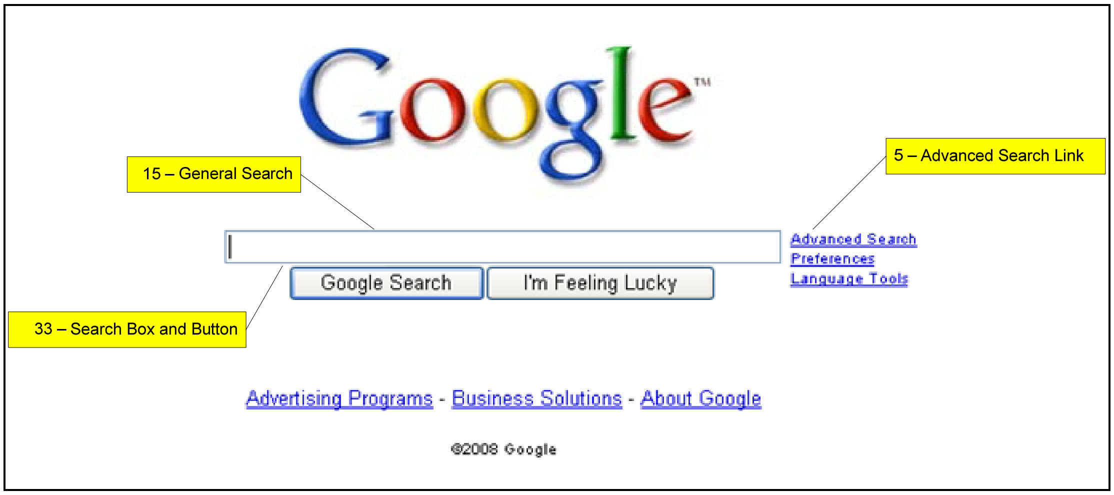


Figure 18. Patterns in aIRPLane identified in Google's general search interface

We then visited the help section of the system and again noted the existing patterns. We then moved to the advanced search interface, noted the existing patterns, and again executed a query. We then returned to the main search interface and looked at the preferences and help available through the main search interface. This was done for the sake of completeness because some patterns occur in different places in different systems. For example in web search engines the pattern *Number of results to display per page* appears under preferences whereas in most IR systems accessed through Drexel's Library the *Number of results to display per page* pattern appears in the results page.

This same process was repeated until the patterns were identified in all 30 IR systems and five web search engines. This resulted in tables that contain a list of the patterns in the first column and the list of systems in the first row (one Table was created for the IR systems and another Table was created for the search engines). If the pattern occurred in the system a 1 was placed in the corresponding cell. If the pattern did not exist a 0 was placed in the corresponding cell. After this was finished, the numbers in a particular row were added together to get the total number of times a particular pattern appeared in all the systems examined. This was done for all rows. This number was also converted to a percentage to examine the percentage of systems that used a particular pattern (Appendix Q). Figure 19 shows the patterns in aIRPLane that appeared in the IR systems, and Figure 20 shows the patterns in aIRPLane that appeared in web search engines. The network diagram of aIRPLane was explained in detail in Chapter 4.

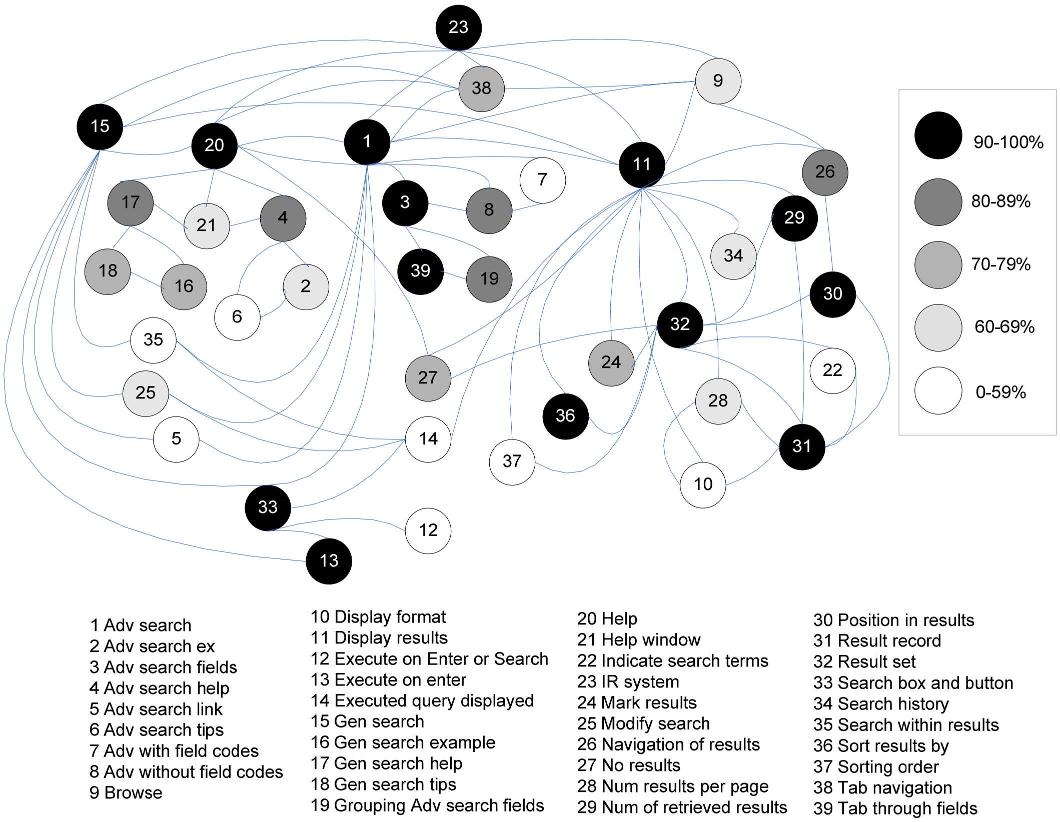
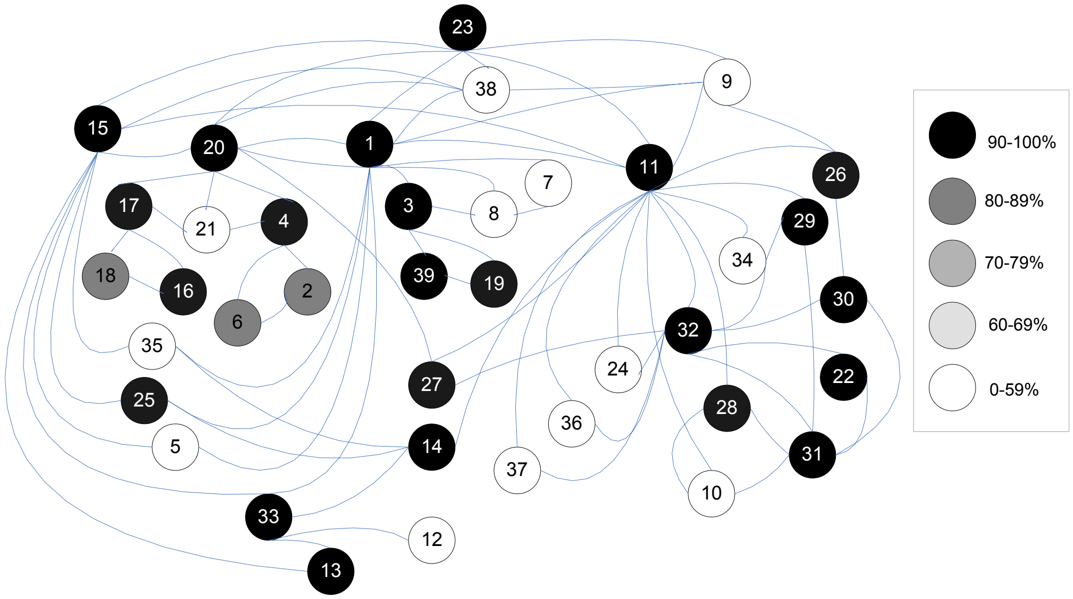


Figure 19. Patterns in aIRPLane identified in information retrieval systems

In examining the network we can see that approximately 35% of the patterns (the darkest colored patterns) were used in more than 90% of the examined information retrieval systems.

Approximately 79% of the patterns in aIRPLane were identified in more than 60% of the 30 systems examined. This network diagram in essence shows us, for the information retrieval systems we examined, the patterns that were identified most frequently; in other words, those patterns that were incorporated into the systems' design most frequently, and those that were identified less frequently, or incorporated into the systems' designs less frequently.



1 Adv search
2 Adv search ex
3 Adv search fields
4 Adv search help
5 Adv search link
6 Adv search tips
7 Adv with field codes
8 Adv without field codes
9 Browse

10 Display format
11 Display results
12 Execute on Enter or Search
13 Execute on enter
14 Executed query displayed
15 Gen search
16 Gen search example
17 Gen search help
18 Gen search tips
19 Grouping Adv search fields

20 Help
21 Help window
22 Indicate search terms
23 IR system
24 Mark results
25 Modify search
26 Navigation of results
27 No results
28 Num results per page
29 Num of retrieved results

30 Position in results
31 Result record
32 Result set
33 Search box and button
34 Search history
35 Search within results
36 Sort results by
37 Sorting order
38 Tab navigation
39 Tab through fields

Figure 20. Patterns in aIRPLane identified in Search Engines

In comparing both network diagrams we see that there is some difference between those patterns identified in traditional information retrieval systems and those identified in web search engines. As compared with the patterns identified in information retrieval systems, we see here that approximately 59% of the patterns in aIRPLane were identified in more than 90% of the web search engines. As illustrated by these two previous figures, the patterns in aIRPLane exist in many popular web search engines and IR systems.

6.3 Evaluating the Usability of IR Systems and Web Search Engines

In an effort to establish a relationship between usability and pattern identification in the IR systems and web search engines, two experts rated the usability of 10 systems. These two experts were not the same two experts who conducted the previous expert review. These two experts were fourth and fifth year doctoral candidates. Both were conducting research in the field of HCI. The experts were asked to perform an expert review and assign a subjective rating, using a 7-point likert-type agreement scale, for overall ease of use, to each of the systems. The experts were given an evaluation sheet with instructions (Appendix R). They were asked to perform three tasks on each system, a general search, an advanced search, and finding the help documentation. After performing these tasks on all systems, the experts were asked to assign a subjective rating, for the overall ease of use, to each of the 10 systems, using a 7-point likert-type agreement scale.

Intraclass correlation was computed using SPSS. The two-way mixed model intraclass correlation coefficient for average measures = .65 indicating a moderate level of agreement (Cronbach’s alpha is .65). The ratings of the two evaluators were averaged and used in further analyses. In an effort to investigate the relationship between the number of patterns in the systems and the overall ease of use rating, Pearson’s correlation was computed. There was a significant positive correlation between the overall ease of use rating and the number of patterns present in the IR systems and search engines ($r = .71$, $n = 10$, $p = .02$), with higher overall ease of use ratings associated with a higher number of patterns.

6.4 Identifying Patterns in Participants' Interfaces

We examined the participants' interfaces to identify which patterns existed in the designs. The patterns that exist in these interfaces were noted on the pattern checklist which is simply a list of the patterns in aIRPLane organized in a fashion that resembles the overall organization of aIRPLane. As patterns were identified they were noted on the pattern checklist. We made one pass through all the participants' interfaces and identified the patterns in each participant's interface. After one week we revisited all the participants' interfaces and again identified the patterns. This was done in an effort to be sure that no patterns were missed. The two evaluation sheets for each participant's interfaces were combined into one evaluation sheet for each participant and the identified patterns were noted and counted for each participant's interface.

After the patterns were identified in all the participants' interfaces, one table was created to compare the number of patterns which were identified in the participants' interfaces for the different conditions (Appendix Q). The table contained the participant's number in the first column and the 39 pattern names in the first row. For each participant the patterns identified in their interfaces were noted in the corresponding cells. The total number of patterns used by each participant was then calculated. After the table was filled in the participants' condition numbers were added to the table by inserting a new first column.

A one-way between-groups analysis of variance was conducted to explore whether there was a difference in the number of patterns used in the participants' interfaces across the groups. The independent variable was the condition with three levels: patterns, guidelines and control. The dependent variable was the number of patterns identified. The significance level for Levene's test for equal variances = .30, therefore we can assume equal variances. There was a statistically significant difference at the $p < .05$ level in number of patterns used for the three conditions F (2, 49) = 7.10, p = .00. The actual difference in mean scores between the groups was rather large, see Figure 21. The observed power = .92.

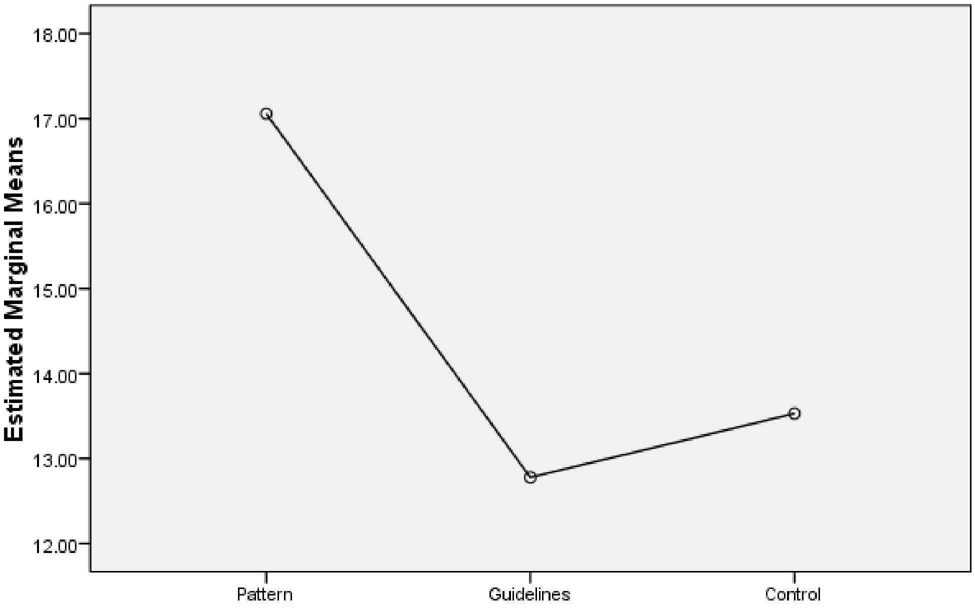


Figure 21. Means Plot for Patterns Used in Participant Designs

Post hoc analyses using Tukey (HSD) post hoc criterion for significance indicated that the number of patterns used was significantly higher in the patterns group (M = 17.06, SD = 4.28) than in both the guidelines group (M = 12.78, SD = 3.06) and control group (M = 13.53, SD = 3.26) (see appendix Q). These analyses suggest that the participants in all conditions used the patterns in aIRPLane, those in the patterns group more than the other groups.

In an effort to investigate the relationship between the number of patterns in the systems and the quality ratings assigned by the experts, Pearson’s correlation was computed. There was a significant positive correlation between the number of patterns present in the participant’s interfaces and the ease of use rating (r = .48, n = 51, p = .01), with higher overall ease of use ratings associated with a higher number of patterns. There was a significant positive correlation between the number of patterns present in the participant’s interfaces and the other three quality ratings, with higher quality ratings associated with a higher number of patterns (level of detail, r = .55, n = 51, p = .01), (completeness, r = .50, n = 51, p = .01), (overall quality, r = .54, n = 51, p = .01). Although there were no statistically significant differences in the quality ratings for the different conditions the correlation suggests that higher quality ratings are associated with a higher numbers of patterns used in the interfaces.

In addition to exploring the differences in the number of patterns used in the participant’s designs we also wanted to examine which of the 39 patterns were used by the participants in the different conditions. The table described above was duplicated and sorted by condition number. This table was broken into three tables, one for each group. The total number of times a particular pattern was used was then totaled for each condition. This resulted in one number that represented the total number of times a particular pattern was used in each condition. Another table was created that contained the pattern names in the first column and the condition names in the first row. The totals from the previous table were entered in the corresponding cells. Percentages were then added to identify the percentage of participants in a particular group that used a particular pattern. The results of these tables were transferred to the network maps for readability. This produced a network map for each condition. These can be seen in the Figures 22-24.

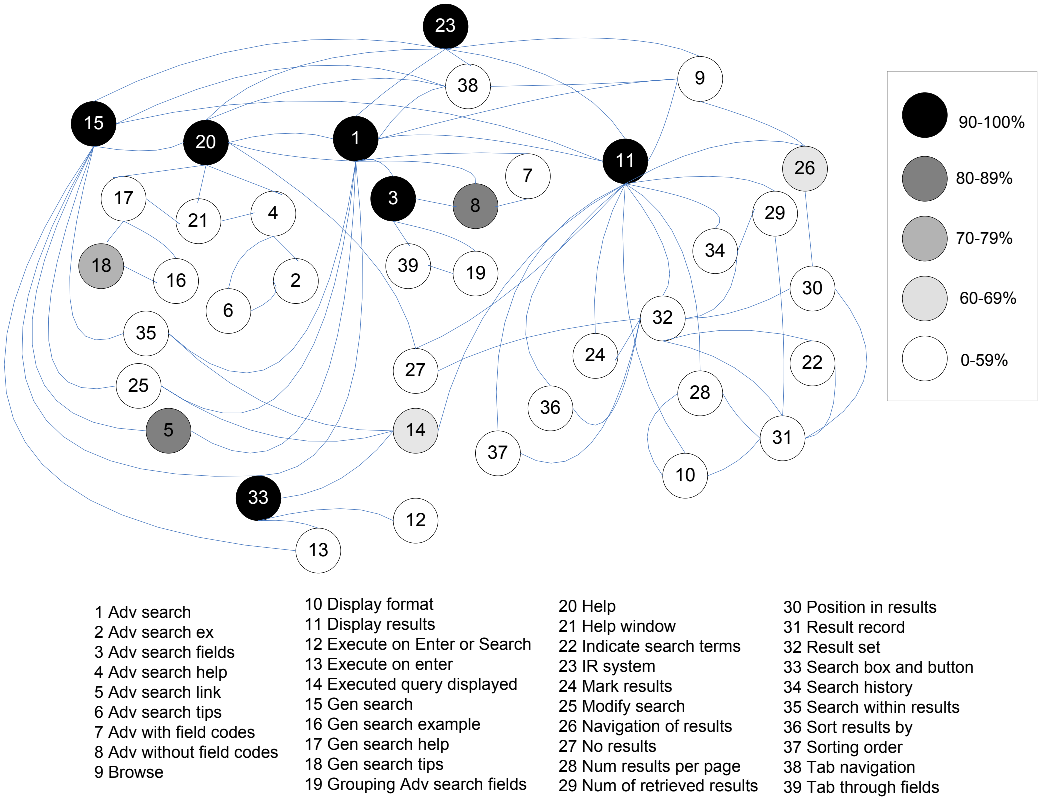


Figure 22. Patterns Identified in Patterns Condition Interfaces

By comparing the five network maps, we can see which patterns were identified in all the participants' designs and those which were identified in the IR systems and search engines examined. Looking at the darker nodes (patterns) in the network diagrams, we can see those patterns that were commonly used by more than 90% of the participants across all the groups.

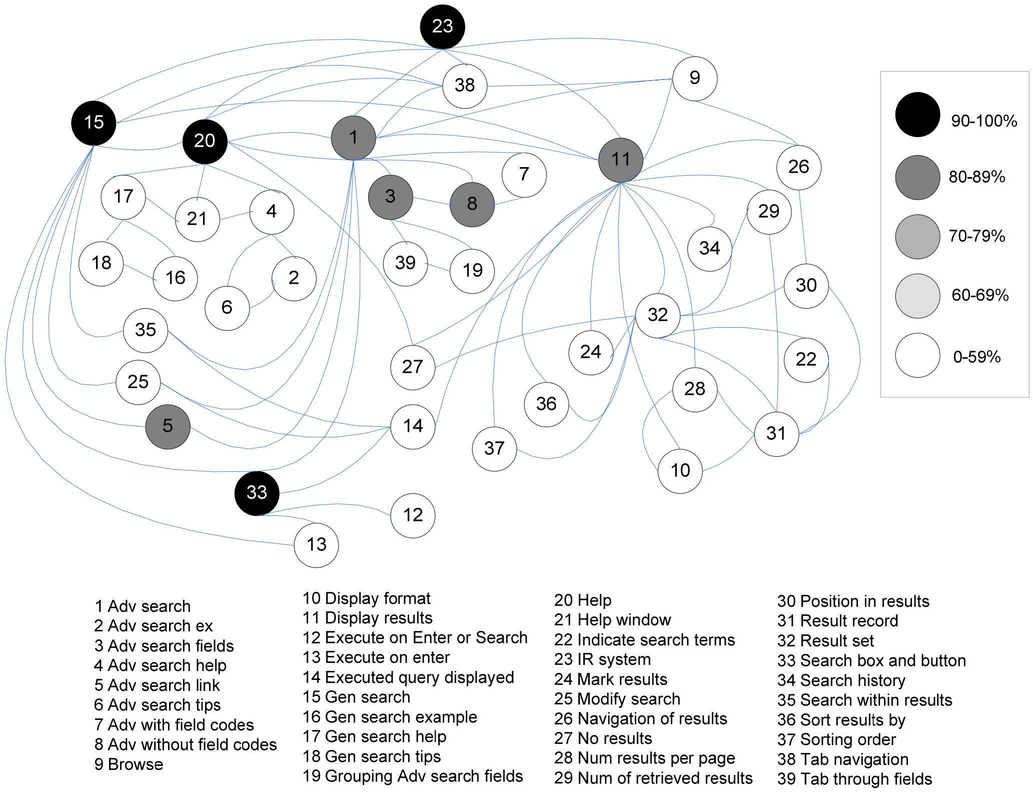


Figure 23. Patterns Present in Guidelines Condition Interfaces

These patterns include: *1 – advanced search*, *11 – display results*, *15 – general search*, *20 – help*, *23 – IR system*, and *33 – result set*. If we examine those used by more than 80% of the participants in the guidelines and patterns conditions, we can add patterns *3 – advanced search fields*, *5 – advanced search link*, and *8 – advanced search without field codes* (these were used in more than 70% of the designs in the control condition). It is important to remember that the participants were asked to include certain functionality in their designs, including things similar to patterns: *15 – general search*, *1 – advanced search*, *20 – help*, and

33 – display results which could account for why these patterns were included in the participants' designs.

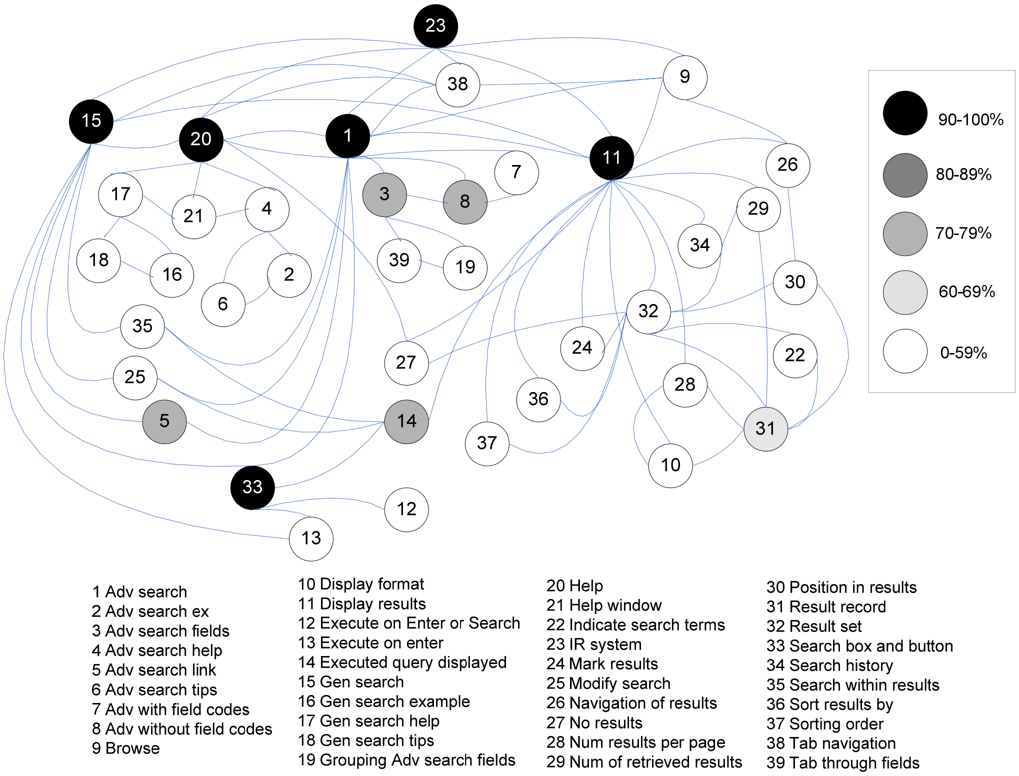


Figure 24. Patterns Present in Control Condition Interfaces

Looking back at the patterns identified in the IR systems and search engines, we see that there are 14 patterns that were identified in more than 90% of the systems. And of these 14 patterns that were often identified in the IR systems and search engines, over 90% the participants in all three conditions used six of these 14 patterns.

6.5 Summary of further exploration

As a result of this exploration we have identified the patterns in aIRPLane used in the information retrieval systems and search engines examined. Approximately 35% of the patterns in aIRPLane were used in more than 90% of the examined information retrieval systems. As compared with the patterns identified in information retrieval systems, we found that approximately 59% of the patterns in aIRPLane were identified in more than 90% of the web search engines. In addition we found a significant positive correlation between the overall ease of use rating for 10 systems and the number of patterns present in those IR systems and search engines (r = .71, n = 10, p = .02), with higher overall ease of use ratings associated with a higher number of patterns. Looking back at the patterns identified in the IR systems and search engines we see that there are 14 patterns that were identified in more than 90% of the systems. And of these 14 patterns that were often identified in the IR systems and search engines, over 90% the participants in all three conditions used six of these 14 patterns.

There was a statistically significant difference at the p < .05 level in number of patterns used by the three conditions F (2, 49) = 7.10, p = .00. The actual difference in mean scores between the groups was rather large. There was also a significant positive correlation between the number of patterns present in the participant's interfaces and the quality ratings, with higher quality ratings associated with a higher number of patterns. The results of these analyses suggest that all the participants used the patterns in aIRPLane irrespective of their condition.

Another interesting thing to point out is the correlation between the number of patterns identified in the IR systems and search engines and the overall ease of use rating as judged by the experts. This correlation is higher in the previous instance described when compared with the correlation between the number of patterns present in the participants' interfaces and the four quality elements (ease of use, level of detail, completeness, and overall quality). We suspect that the differences in actually interacting with a system and looking at a prototype, in which one needs to assume certain things about functionality and interactions, may be the reason for the higher correlation in the case where the experts actually used the systems.

7. DISCUSSION

For decades patterns and pattern languages have been discussed in HCI literature. In much of the literature the focus has been on the possible benefits of using patterns. It has been suggested that patterns may support the reuse of solutions to common design problems (Erickson, 2000; Granlund et al., 2001). In the software engineering community it seems that reuse is the goal of design patterns (Fowler, 1997; Gamma et al., 1995). In HCI the focus has not only been on reuse but, also on facilitating communication within design teams and between designers and non-designers. It has been suggested that communication may be facilitated by providing a common language, or a lingua franca (Erickson, 2000), that both designers and non-designers could use to communicate. It has also been suggested that pattern languages may aid in the capture and sharing of design knowledge (Dearden & Finlay, 2006), in addition to supporting both design and evaluation activities.

While much of the literature has focused on the promise of pattern languages (Pemberton, 2000), or the possible benefits they provide, there has been little empirical work to support these claims (Dearden & Finlay, 2006). The results of this study indicate that a pattern language had no apparent affect on the quality of the information retrieval interfaces designed by the participants or the time to design the interfaces. The results of this empirical study suggest that the value of pattern languages in HCI may not be in reuse at the early stages of design, in a domain that designers are familiar with.

In this discussion, we would like to explore how pattern languages support comprehension, communication, problem solving, and knowledge management. Although we do not have empirical support for this at this time, we review a number of instances which point to pattern-like structures in comprehension, communication, problem solving and knowledge management literature.

Before addressing these issues we would like to discuss the results of this study as they relate to our research questions and the other empirical studies that have been conducted. In examining the impact of aIRPLane on the design of information retrieval interfaces three primary research questions were considered:

RQ 1. What is the impact of a pattern language on the quality of the designed interfaces?

RQ 2. What is the impact of a pattern language on the time it takes to design interfaces?

RQ 3. How do the participants view the underlying relationships among patterns?

In this Chapter the results of this study are discussed as they relate to the research questions above. The majority of the discussion surrounds research question 1. The results are described in comparison to the results of the other empirical studies in this area. As described in the previous Chapter we found that the exposure to aIRPLane did not have a statistically significant affect on the quality of the designed interfaces or the time to design the interfaces. This is consistent with other empirical studies in this area (Chung et al., 2004; Saponas et al., 2006). After providing a summary of the results, we explore why we may, and may not, expect a pattern language to have an affect on the quality of the interfaces designed.

7.1 Summary of Results

In this study aIRPLane did not have an apparent impact on the quality of the designed interfaces. This is consistent with the findings of other studies in this area (Chung et al., 2004; Saponas et al., 2006). There were no statistically significant differences found between the groups for the quality of the interfaces (as measured by ease of use, completeness, level of detail, and overall quality). However, as described in the previous Chapter, it seems that the patterns were used by participants from all three conditions. In addition, it seems that pattern usage in IR systems and search engines, is highly correlated with overall ease of use ratings as judged by two experts, as described in the previous Chapter.

It only seems fitting then to ask, if patterns seem to have no impact on the quality of the first iteration of a design…what is the value of patterns in HCI? This question will be the focus of our discussion after we address the other 2 research questions.

Related Research

In the two previous controlled studies (Chung et al., 2004; Saponas et al., 2006) that examined the impact of patterns on the quality of a design, the authors concluded generally, that there was no difference between the control groups and the groups that used pre-patterns (see literature review and Saponas et al. for a discussion of pre-patterns). The exception to this is the detail score in Saponas et al. which was on average higher for the control condition, not the patterns condition. The participants in both Chung et al. and Saponas et al. were pairs of designers with varying levels of experience. The participants in this study were individual participants (both undergraduate and graduate students with varying levels of coursework and work experience). Although the participants were not from the same population, and the design task was focused on a different domain, our results are consistent with the results of previous work in this area (Chung et al., 2004; Saponas et al., 2006).

Golden, John, & Bass (2005a; 2005b) have conducted controlled studies involving the use of what the authors refer to as USAPs or Usability Supporting Architectural Patterns, the results of their studies suggest that the use of USAPs increased the quality of the participants' solutions. Although the results are interesting the patterns are not interaction patterns as we tend to refer to them in HCI. They differ in terms of the elements they contain and their purpose (see Chapter 2 for discussion). Therefore it is not easy to generalize their results to interaction patterns.

Participant feedback

In related work such as Chung et al. (2004) the authors report that participants rated the patterns as relatively useful for the design task (3.6 out of 5). Here, the participants in both the guidelines condition (average 3.83 out of 5) and the patterns condition (average 4.35 out of 5) found the respective technique relatively useful for the design task (1 = strongly disagree, 5 = strongly agree).

Time to Design

In this study the structuring technique had no apparent impact on the time to design the interfaces. Previous studies (Chung et al., 2004; Saponas et al., 2006) did not report whether the technique had an impact on the time taken to design the interfaces. Saponas et al. point out that those in the patterns group spent on average 26 minutes reading the patterns which accounted for 11% of their total time. Because it is not clear to us whether the rest of the total time was actually spent designing, it is difficult to compare the results. In this study we also found a significant positive correlation between the time taken to design the interfaces and the four quality elements (see Chapter 5).

7.2 Promises, Promises

Here, we would like to borrow Lyn Pemberton’s title and explore *The Promise of Patterns Languages for Interaction Design*. In other words, we would like to explore why we may expect a pattern language to have an affect on the design of interfaces. One indicator is the amount of literature which has suggested for decades that there are many benefits to using patterns in HCI, and this literature seems to be continually growing. In addition, the empirical studies which have been conducted conclude with positive remarks about patterns being helpful in design and communication. There is much enthusiasm surrounding patterns, but there is little empirical support demonstrating the value of pattern languages in HCI, as previously described in Chapter 2.

We could possibly expect an impact if, as suggested by Chung et al. (2004), the participants were new to a domain, or if we were examining the quality of later design iterations, possibly a third or fourth iteration of a design. In the next section we explore why we would not expect a pattern language to have any apparent affect on the design of interfaces.

Revisiting the Promises

We would like to explore why a pattern language may *not* appear to have an affect on the design of first iteration interfaces. At first it seems somewhat surprising that the results of this study suggest that a pattern language had little or no apparent affect on the design of information retrieval interfaces, especially considering the claims made in the literature about how patterns may be useful design tools. But, if we revisit the empirical studies that have been conducted in this area, (Chung et al., 2004; Saponas et al., 2006) they too found practically no positive affect on the quality of the designs or the time to design the interfaces due to the patterns. In both Chung et al. and Saponas et al. a patterns condition was compared to a control condition. And in both studies there was practically no difference between the groups in terms of the quality of their designs. So the obvious question to follow is: Why is it that there is no affect of the patterns on the quality of the designs? We explore this a bit here.

As suggested by Chung et al. (2004) if we were examining the quality of later design iterations, possibly a third or fourth iteration of a design we may expect a pattern language to have an impact. In this study and both previously conducted controlled experiments (Chung et al., 2004; Saponas et al., 2006) the designers were creating first iterations. First iterations of a design are exactly that, a first try, the first time anything is put on paper, therefore there is nothing already existing to comment on, or discuss, or evaluate. First iterations are difficult because there are usually many choices that need to be made, from many alternatives, and there are consequences that need to be considered for each decision. This is a difficult task that requires a lot of cognitive processing. We know from work in design studies that our cognitive capacity is limited and that it is not possible for designers to comprehend and consider every alternative (Alexander, 1964; Brooks, 1987; Rittel & Webber, 1984). It seems possible that our ability to process the problem space and a design technique at the same time could be too much to process. We should also recognize that in design studies there is a notion of a designer's repertoire and it is believed that designers tend to rely on their design repertoire when designing (Schön, 1987).

In addition, it is fairly safe to assume, although it was not asked, that all the participants, in this study, have used some type of information retrieval system before. And as we know the patterns in aIRPLane exist in many systems (as described in detail in the previous Chapter), systems that these participants have most likely been in contact with. If they have not been in contact with the traditional IR systems, it is fairly safe to assume that most, if not all, participants have used Google or a similar web search engine. As noted in the last Chapter, 28 of the 39 patterns in aIRPLane were identified in Google. It is therefore safe to assume that the participants in this study have been in contact with some of these patterns in the past.

Following on this point, it seems reasonable that the participants relied on their knowledge and experiences using IR systems to help them design the IR interfaces in this experiment. We would like to acknowledge that this is an aspect of this study that is unlike other empirical studies in this area. In both Chung et al. (2004) and Saponas et al. (2006) some of the participants were not familiar with the domain in which they were asked to create a design. As suggested by Saponas et al. if the participants in this study were asked to design interfaces for a system with which they were not familiar, the results of this study may have been different. But even in previous studies, such as Chung et al. and Saponas et al., in which not all designers were familiar with the domain in which they were designing, the designers stated that they believed that patterns would be useful to those who are not familiar with a domain. Yet the results to support this claim are anecdotal, not quantitative measures.

In answer to the question posed above, it could be that there was no affect of the patterns on the quality of the designs because all the participants were in fact using patterns. Is it very possible that the participants were using the patterns they were familiar with, and had been in contact with previously through their use of IR systems and search engines, to help them design the IR interfaces.

In an attempt to further illustrate this point we would like to introduce a simple thought experiment here. Suppose we gave a number of participants the task of designing the layout of a house. Let's imagine they were asked to design a one story house with three bedrooms, two bathrooms, a kitchen, living room, and dining room. Now, let's suppose we gave half of the participants some patterns from Alexander's pattern language (1977) (or any other architectural pattern language) and the other half were left to design the house layout without any structuring technique. It is reasonable to assume that we would not expect the participants in the two groups to design drastically different layouts for the house.

We would not be surprised if the layouts looked something like those pictured in Figure 25. We would not be greatly surprised if everyone included some type of foyer or entranceway into the house. We would not be incredibly surprised if they all placed the kitchen and dining room next to each other.

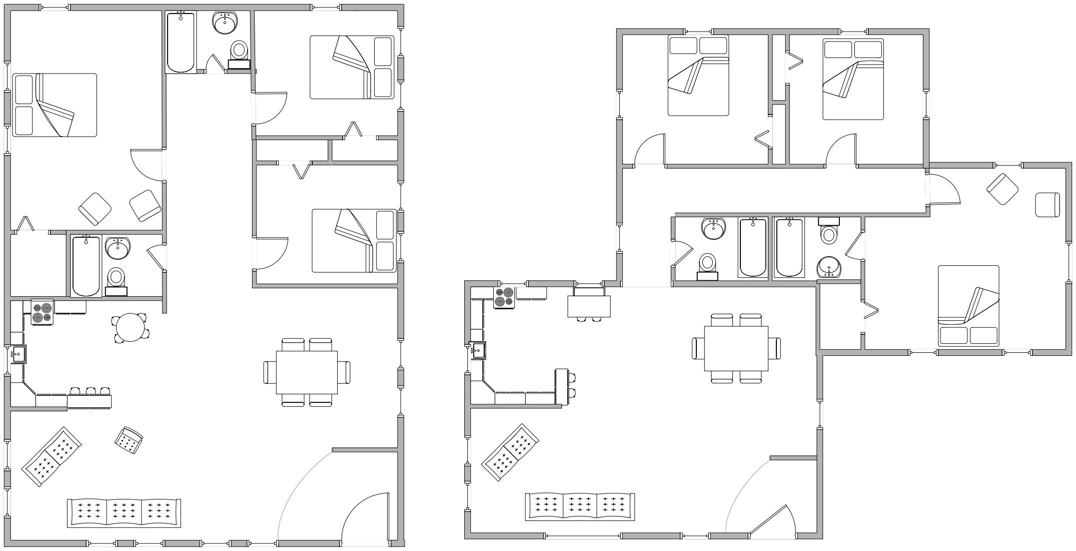


Figure 25. Examples of Two Possible House Designs

We would not be very surprised if the bedrooms were placed near the back of the house away from the common areas. It does not seem that any of this would be of great surprise. But, what may be a surprise to some is all of these things just described are patterns, patterns that are commonly used because they are successful solutions to problems in the design of single family homes. Figure 26 shows a possible design of a single family home with some of Alexander’s patterns identified throughout the house.

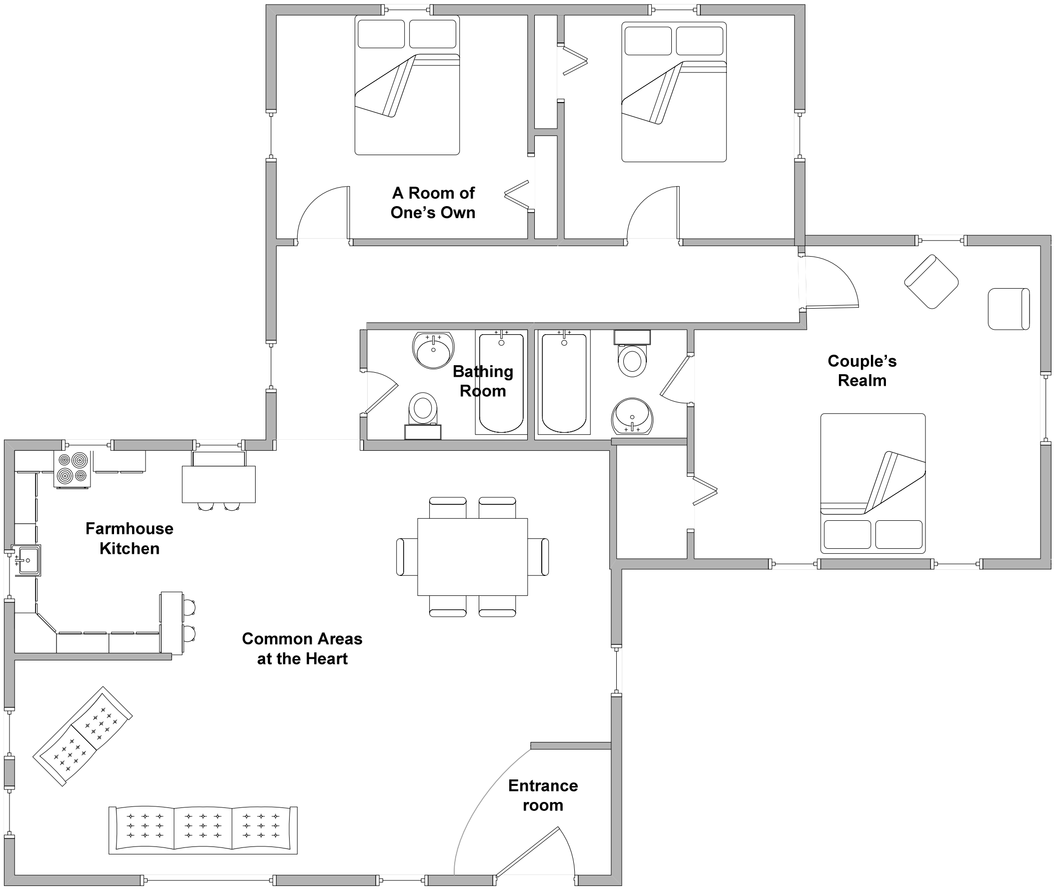


Figure 26. House Design with Alexander's Patterns Identified

From this simple thought experiment it seems that we should not question why there was no affect of the patterns on the quality of the designs produced. It seems rather reasonable (although we do not have empirical evidence at this time to support this) to suggest that the participants used the patterns that they had been in contact with while using information retrieval systems in the past. Although the participants may not be aware that they were using "patterns" or refer to them as patterns, they may have been actually using patterns or a pattern-like structure.

We would like to propose another thought experiment of a different kind here. Suppose we asked a number of participants to write a fairy tale. Let us assume that the participants are all from the United States. Now suppose that we gave half of the participants the schema for a fairy tale and let's suppose the other half of the participants were not given any structuring technique. Schemata are structures used to understand events, stories, empirical HCI papers, and so on (they will be discussed in further detail in a later section). We would not expect the participants in the two groups to write drastically different stories.

It does not seem that we would be greatly surprised if all the participants started their story with "Once upon a time…" and ended it with "and they lived happily ever after." We probably would not be surprised if all the stories contained "good guys" and "bad guys" and some sort of conflict. We are not surprised by this because we are all familiar with this existing story schema. And the fact that we are not surprised points to the fact that these things do exist and they are real.

7.3 What is the value of patterns?

Now that we have explored *why* it may be that there was no apparent affect of a pattern language on the design of the interfaces, let's explore what this may suggest. In other words, what is the value of patterns? We suggest that we, as a community, begin to identify, through empirical means, the benefits of using pattern languages in HCI. There have been many useful contributions to the literature that do not have an empirical component, but we urge the community to slow down, take a step back, and evaluate our current state, just as Dearden & Finlay (2006) did in their review, so that we as a community can continue to move forward in this area by means of empirical studies.

We have discovered that a pattern language may not improve the quality of a design, in the first iteration, when designers are familiar with the domain. Other promises for pattern languages include: they may be useful as an evaluation technique, they may aid in the capture and sharing of design knowledge, and they may help improve communication between those involved in the design process. Here we would like to explore the value of patterns as communication tools, as boundary objects, as a lingua franca or a common language that people may use to communicate.

7.4 Patterns as Communication Tools in HCI

The intention of a pattern language, as described by Alexander (1977; 1979), was to capture the heart of successful solutions to recurring design problems in architecture and provide a common language that architects and non-architects could both use to communicate. Alexander's goal in documenting these patterns was to capture what he referred to as the quality without a name, a quality which made places and the people in them feel alive and whole. Much of the focus in HCI has been on the reuse of patterns as design tools but, patterns have also been proposed as communication tools. These claims, like most others about patterns, have not yet been evaluated.

As previously discussed, design involves a lot of communication between people from many different disciplines, and many times symmetry of ignorance (Rittel & Webber, 1984), or the problem in effectively communicating with people from different disciplines, is one of the most difficult and common problems in design. Authors in HCI, specifically Erickson (2000) have discussed using pattern languages as a *lingua franca*, or a common language which allows designers and non-designers to communicate with one another. Erickson describes the idea of a lingua franca as a common language which is accessible to all participants involved in a design process. Bayle et al. (1998) report on the results of a workshop at CHI 97 which focused on the use of patterns and pattern languages in interaction design. Bayle et al. point to a number of properties which pattern languages have that might enable them to serve as lingua franca for the diverse community of interaction designers including:

- they are based on concrete prototypes
- they work at multiple levels (community, group, individual) and they try to tie the levels together
- they attempt to bridge the gap between the physical and social worlds
- they are amenable to gradual development (p. 18)

In discussing the communication problem that occurs in design between those from different disciplines, Borchers (2001) points out that

> If this communication fails, the result is that the methods, paradigms, and ultimately the values of each profession are not understood, and consequently cannot be respected, by the other disciplines. Any method that simplifies this mutual understanding would benefit the design process, and the resulting product (p. 5).

Borchers suggests that pattern languages may be a way of facilitating communication in design.

We, as a community, seem to have focused too much on the outcome of the design process, and reuse, when in fact we should probably be focusing on the design process itself and the communication that takes place throughout the design process. We suggest that we, as a community, shift our focus to the design process and the communication that takes place within this process. Although there has been little empirical work surrounding patterns as communication tools, there seems to be some evidence that suggests patterns may be useful communication tools. In the following section, we provide a brief review of relevant literature which may suggest patterns could be useful in communication. We also identify the similarities between patterns and other communication tools.

7.5 Aren’t We Already Using Patterns to Communicate?

It is no secret that communication is a major part of design and at times a major problem in design (Borchers, 2001; Kim, 1990; Rittel & Webber, 1984). Erickson (2000) describes design as a communicative process and he points to the tools we use in design that aim to improve communication as evidence, for example, storytelling, scenario-making, prototype building, and user testing. In this section we review a few well-known works in design literature which address the communication process in design with the intention of illustrating how the communications described therein, are pattern-like. We review a number of ideas including reflection-in-action, war stories, and schemas.

7.6 Patterns as Reflective Discussion

In *Educating the Reflective Practitioner,* Schön (1987) describes a design review, essentially the reflective discussion that occurs, between an instructor, Quist, and a student, Petra, in an architectural design studio. (Please note this is an actual communication that took place in a real design studio, see Schön, 1987, p. 44.) We refer the reader to this passage for the details of the communication. We only include portions of the communication here. The back and forth communication that occurs between the instructor and the student in an architecture design studio seems to exhibit the essence of a pattern (problem, context, solution). Schön (1987) describes a situation in which the student, Petra, shows the instructor her sketches and states the **problems** she is having.

> Petra: I am having trouble getting past the diagrammatic phase – I’ve written down the problems on this list. I’ve tried to butt the shape of the building into the contours of the land there – but the shape doesn’t fit into the slope. (p. 46)

As the student describes her problems to the instructor, the instructor asks some questions that require a response from the student. In these questions and answers the **context** of the problem is discussed.

> Q: Is this to scale?
> P: Yes.
> Q: Okay, say we have introduced scale. But in the new setup, what about north-south?...
> P: This is the road coming in here, and I figured the turning circle would be somewhat here- (Schön, 1987, p. 48)

The instructor and the student continue with the conversation and also create some **drawings** as they are talking. The conversation continues with the instructor exploring the consequences and implications of some proposed actions, or in other words, the **forces** at play.

> Q: Now you would give preference to that as a precinct which opens out into here and into here, and then of course, we'd have a wall – on the inside there could be a wall or steps to relate in downward. Well, that either happens here or here, and you'll have to investigate which way it should or can go. If it happens this way, the gallery is northwards – but I think the gallery might be a kind of garden – a sort of soft back area to these. The kindergarden might go over here – which might indicate that the administration over here – just sort of like what you have here – then this works slightly with the contours (Schön, 1987, p. 52)

Schön (1987) describes this type of activity as "a 'what if' to be adopted in order to discover its consequences" (p. 57). There is clearly a similarity between what Schön is describing here and what Alexander describes as forces. The instructor and the student then seem to find a **solution**, together.

> P: Where I was hung up was with the original shape; this here makes much more sense.
> Q: Much more sense – so that what you have in gross terms is this…Now you have to think about the size of this middle area. You should have the administration over here.
> P: Well, that does sort of solve the problems I had with the administration blocking access to the gym.

> Q: No good – horrible, it just ruins the whole idea – but if you move it over there, it is in a better location and it opens up the space (Schön, 1987, pp. 53-54)

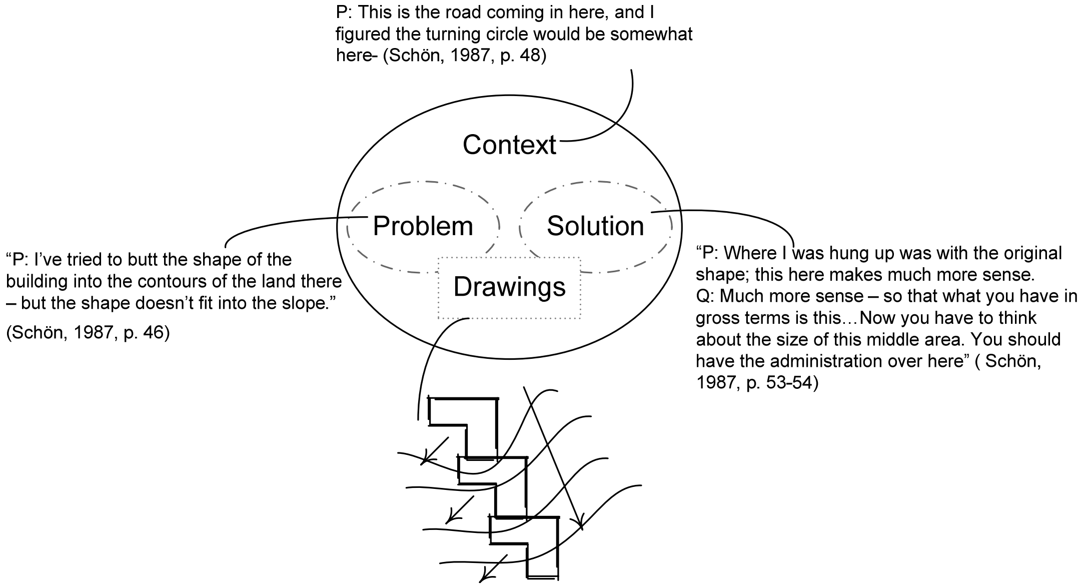
To review, we have identified **problems**, **context**, **drawings**, **forces**, and **solutions** in the conversation that takes place between the instructor and student in a design studio. Schön (1987) provides his thoughts on what is taking place, he describes:

> What I want to propose is this: Quist has built up a repertoire of examples, images, understandings, and actions. His repertoire ranges across the design domains. It includes sites he has seen, buildings he has known, design problems he has encountered, and solutions he has devised…When a practitioner makes sense of a situation that he perceives to be unique, he sees it as something already present in his repertoire. To see *this* site as *that* one is not to subsume the first under a familiar category or rule. It is rather to see the unfamiliar situation as both similar to and different from the familiar one, without at first being able to say similar or different with respect to what…Seeing *this* situation as *that* one, a practitioner may also do in this situation as in that one…Indeed this process of *seeing-as* and *doing-as* may proceed without conscious articulation (pp. 66-67) (italics in original).

We would like to point out not only the obvious similarities between patterns and the conversation that is taking place in the design studio but, also the obvious similarities between patterns and what Schön is describing as being contained in the designer's repertoire.

The difference between the communication described therein and a pattern is the way in which the information is communicated. In the case of the instructor and the student, they are communicating verbally, face to face, and through drawings. In the case of interaction patterns, as we see them in HCI today, the material is documented on paper or websites so that another person can retrieve it when necessary.

If we consider the essence of a pattern, we must include a **problem** situated in a **context**, and a **solution**. Drawings are also included many times. This is the essence, or heart, of a pattern. If we consider these elements the essence of a pattern, we can discuss what a pattern is and what a pattern is not. In the conversation between the instructor and student, as described by Schön, we see all of these elements, as illustrated in Figure 27.



P: This is the road coming in here, and I figured the turning circle would be somewhat here- (Schön, 1987, p. 48)

"P: I've tried to butt the shape of the building into the contours of the land there – but the shape doesn't fit into the slope."
(Schön, 1987, p. 46)

"P: Where I was hung up was with the original shape; this here makes much more sense.
Q: Much more sense – so that what you have in gross terms is this…Now you have to think about the size of this middle area. You should have the administration over here" (Schön, 1987, p. 53-54)

Figure 27. The Essence of a Pattern as Reflective Discussion

The conversations between the instructor and the student, described by Schön, seem to contain the essence of a pattern, although they are not contained or documented in the same form.

7.7 Patterns as Knowledge Management

In *Talking about Machines: An Ethnography of a Modern Job*, Julian Orr (1996) portrays the practices of experienced and skilled photocopier technicians. Orr describes that narratives or stories, later deemed war stories, form a "primary element of this practice" (p. 2). We describe the technicians' use of war stories here and provide an example but, we refer the reader to the text for further discussion of war stories. Orr describes that the technicians swapped war stories in a number of situations and settings and for a number of reasons, but the one thing that seems to remain constant throughout is the structure of the stories. Orr (1996) describes,

> The use of war stories is a prominent feature of diagnosis among the technicians. These stories are anecdotes of experience, told with as much of the **context** and technical detail as seems appropriate to the situation of their telling. At a minimum they name the technician doing the work, the machine to which it has been done, the **problem**, and its **solution**; in the majority of cases I observed, the technician telling the story is the one to whom it happened (bold added for emphasis) (p. 125).

In this description we see the main elements of a pattern: context, problem, solution. Orr describes many of the benefits of war stories that others have suggested patterns have, such as reuse and the capture and sharing of design knowledge.

> Once war stories have been told, the stories are artifacts to circulate and preserve. Through them, experience becomes reproducible and reusable. At the same time, each retelling is, in a sense a re-representation. The stories originate in problematic situations and are told or retold in diagnosis when the activity they represent becomes problematic again. They are retold in the consideration of a present problem, when the issue of comparability of context with some previous experience has arisen, and this renders the previous, completed episode once more problematic (Orr, 1996, p. 126).

It is clear from this discussion that the stories circulate and that through the sharing of the stories the technicians learn from one another. Orr (1996) describes that almost everyone in the corporation knew that the technicians told war stories but, the views or attitudes towards war stories varied. For example Orr describes, “technicians told me that their immediate managers delayed starting team meetings because the technicians were telling each other stories about their most recent experiences, and this information exchange was perceived as useful or even vital” (p. 140).

The following quote from one of the technicians suggests that people other than the technicians did see value in the telling of the stories, “the weekly team meeting is supposed to help these bits of information circulate. Frank says that John, the team manager, always starts the meeting late because the team members are sitting there talking about all their problems with the machines, teaching each other what they learned" (Orr, 1996, p. 51). Orr summarizes the importance of the war stories in stating, "technician's stories *are* work; they are part of diagnosis, and they help preserve the knowledge acquired for the benefit of the community. Stories are more than a celebration of practice; they are an essential part of the practice to be celebrated" (1996, p. 143) (italics in original).

Orr's study of Xerox's copier repair technicians was one factor that led to the development and use of the EUREKA system at Xerox (Bobrow & Whalen, 2002). The EUREKA system was designed over a number of years using a bottom-up participatory approach. The goal of the system was to capture and disseminate tips for repairing copiers. The tips (or pattern-like stories) consisted mainly of a **problem**, **cause**, and **solution** (Bobrow & Whalen, 2002). EUREKA has been successfully used at Xerox in France, Canada, and the United States for many years (Bobrow & Whalen, 2002).

In light of the number of known failures in Knowledge Management (Malhotra, 2002) it would be helpful, for us, as a community, to closely examine how EUREKA was designed, implemented, and used at Xerox. It is obvious that there is a resemblance between patterns and the EUREKA tips that contained problems, causes, and solutions. This is the essence of a pattern. This example clearly illustrates that people do use pattern-like structures to communicate and solve problems in real world organizations. It seems that we can learn a lot from the challenges and success of EUREKA.

7.8 Patterns as Comprehension Tools

Schemas, frames, and scripts are structures used to understand events, stories, empirical HCI papers, and so on. Kintsch (1998) describes that "schemas, frames, and scripts are structures used to coordinate concepts that are part of the same superstructure, or event" (p. 36). Here, Kintsch is suggesting that there may be schemata for different events and stories. Van Dijk (1977), in a more general sense, defines frames as "knowledge representations about the 'world' which enable us to perform such basic cognitive acts as perception, action, and language comprehension" (p. 19). Tannen & Wallant (1993) provide a good review of frames, schemas, scripts and their origins. Tannen & Wallant point to two categories in which these terms are used. They distinguish between interactive frames and knowledge structures. They refer to interactive frames as "a sense of what activity is being engaged in, how the speakers mean what they say…in order to comprehend any utterance, a listener (and a speaker) must know within which frame it is intended: for example, is this joking? Is it fighting?" (Tannen & Wallat, 1993, p. 60). In a broader sense Tannen & Wallant use the term knowledge structure to "refer to participants' expectations about people, objects, events and settings in the world, as distinguished from alignments being negotiated in a particular interaction" (p. 60).

As suggest above in Kintsch's definition, there are different types of schemata, including event schema, scene schema, and story schema. Mandler (1984) describes each of these well. We refer the reader to this text for more information. Here, we will briefly discuss story schema. Story schema, just like schema in general, are mental structures, but in this case specifically about stories. Mandler describes a story schema as "a mental structure consisting of sets of expectations about the way in which stories proceed" (p. 18). Mandler continues by stating a story schema "is a mental reflection of the regularities that the processor has discovered (or constructed) through interacting with stories" (p. 18). This discovery or construction through interaction with stories seems to be the same way we discover and construct patterns, through experience with artifacts in our environment, for example, like our experiences with homes and interactive systems. Mandler also describes the rules for story structure,

> In the kind of relational structure that has been described, units are recognized primarily because of what has gone before (and what comes after); for example, the identical string of words could represent a beginning event, an attempt, or an outcome. The listener makes a determination of the constituent to which the sentence belongs primarily on the basis of its relation to the just preceding and immediately following sentences (1984, p. 25-26).

What Mandler describes is precisely why we are not able to understand things like movies or stories that start at the end, because we are not aware of the right rules or structure for understanding these things. Someone that is not familiar with a particular schema or frame may find it difficult to understand a story that follows that particular schema or frame. In other words a person must understand the schema to fully understand the story. Therefore, if you do not have the appropriate schema you may not recognize that a pattern is a pattern, or understand a pattern when you see one. But, with the introduction of the appropriate schema we are able to communicate these things with one another.

Let’s think back to the thought experiment proposed earlier, asking participants to write a fairy tale. And as suggested earlier, let’s suppose we gave half of the participants the schema for a fairy tale, and the other half were not given any structuring technique. We discussed earlier that we would not be surprised if all the stories, regardless of the group the participants were in, began with “once upon a time” and ended with “and they lived happily ever after.”

On a related note, if we gave the stories the participants had written to another group of participants we would expect that they could understand each of the stories (assuming they are also from the United States). But, let’s suppose now that we gave half of the participants the fairy tales and the other half of the participants *War of the Ghosts,* a North American folk tale used in Bartlett’s (1995) experimental studies. We would probably not expect the participants to understand the *War of the Ghosts* as well as the participants understood the fairy tale. We have included the *War of the Ghosts* here for illustrative purposes.

> One night two young men from Egulac went down to the river to hunt seals and while they were there it became foggy and calm. Then they heard war-cries, and they thought: "Maybe this is a war-party". They escaped to the shore, and hid behind a log. Now canoes came up, and they heard the noise of paddles, and saw one canoe coming up to them. There were five men in the canoe, and they said:
> "What do you think? We wish to take you along. We are going up the river to make war on the people."
> One of the young men said, "I have no arrows."
> "Arrows are in the canoe," they said.
> "I will not go along. I might be killed. My relatives do not know where I have gone. But you," he said, turning to the other, "may go with them."
> So one of the young men went, but the other returned home.
> And the warriors went on up the river to a town on the other side of Kalama. The people came down to the water and they began to fight, and many were killed. But presently the young man heard one of the warriors say, "Quick, let us go home: that Indian has been hit." Now he thought: "Oh, they are ghosts." He did not feel sick, but they said he had been shot.
> So the canoes went back to Egulac and the young man went ashore to his house and made a fire. And he told everybody and said: "Behold I accompanied the ghosts, and we went to fight. Many of our fellows were killed, and many of those who attacked us were killed. They said I was hit, and I did not feel sick."
> He told it all, and then he became quiet. When the sun rose he fell down. Something black came out of his mouth. His face became contorted. The people jumped up and cried.
> He was dead.

As Kintsch (1977) suggests the participants would not be able to comprehend the story unless they had the appropriate schema.

Let’s explore one more thought experiment here before moving on. Let’s suppose we asked a group of participants to write an Alaskan Indian story, as described by Kintsch (1977). And let’s suppose that we gave half of the participants the schema for an Alaskan Indian story and let’s suppose the other half were given no structuring technique. It seems highly unlikely that the participants would be able to produce such a story, unless the participants had encountered the schema for an Alaskan Indian story in the past. An Alaskan Indian story, as described by Kintsch, contains four episodes which are not related in a causal-temporal way. This is very different from the schema that most of us have for a story. But, if the subjects had some past experience with Alaskan Indian story it seems reasonable to assume that they would be able to produce such a story. This suggests that we do use patterns, or schemas, which are pattern-like in communication and comprehension.

7.9 Patterns as Explanation Patterns

An explanation pattern is a structured account of an event or situation that is used to explain similar situations or events. As described by Schank (1986), "an explanation pattern is a fossilized explanation. It functions much in the same way as a script does. When it is activated, it connects a to-be-explained event with an explanation that has been used at some time in the past to explain an event similar to the current event" (p. 110). Patterns too seem to be fossilized explanations in the way Schank describes them but, in the HCI they are simply fossilized explanations of a context, a problem, and a solution for interactive systems design. In the following quote Schank describes the structure of an explanation pattern,

> **An Explanation Pattern consists of multiple parts**. First, we have an **index to the pattern**. This index is made up of a combination of states and events. Second, we have **a set of states of the world under which those indices can be expected to be active**. When those states of the world are achieved, the indices fire. The next part is **the scenario**. The scenario is essentially a little story that is a carefully constructed causal chain of states and events that starts with the premise of achieving the combination of states and events in the index and presents a plan of action for achieving that combination. The fourth part is **the resultant state that follows from the scenario**. This state may also be used as an index initially. Also, **attached to an XP are explanations that have been previously compiled from that XP**. In this way, one can be reminded of similar cases. Thus XPs are themselves a means of traversing memory. Last, **every XP has a reason attached to it that can both serve as an index and as the ultimate explanation behind the use of the explanation embodied in an XP** (pp. 111-112 bold in original).

In this description of the structure of explanation patterns we see a number of similarities with the structure of patterns in HCI. One of the more interesting similarities is Schank’s description of the index to a pattern which seems to have similarities to the relationships between patterns in HCI or the *related patterns* section of some patterns in HCI.

Not only do patterns in HCI and explanation patterns, as described by Schank (1986), seem to have similarities in their structure but, it seems that the way in which they are activated, retrieved from memory, and used may also be similar. In describing how we use explanation patterns, Schank describes,

> The point is that one does not have to compute everything as if it had been seen for the first time. Understanding relies upon our ability to take shortcuts by assuming that what we have just seen is not that different from something with which we were already familiar (1986, p. 109).

It seems reasonable to assume that when a person is asked to design something, like interfaces for an information retrieval system, that they assume that what they are being asked to design is not that different from those systems they are already familiar with and therefore use what they know from experience to help them with the task. Just as we use explanation patterns, it seems reasonable to assume that we also use interaction patterns in design.

7.10 Summary of Discussion

While much of the literature has focused on the promise of pattern languages (Pemberton, 2000), or the possible benefits they provide, there has been little empirical work to support these claims (Dearden & Finlay, 2006). The results of this study indicate that a pattern language had no apparent affect on the quality of the information retrieval interfaces designed by the participants or the time to design the interfaces. Yet we observed that all participants, regardless of the experimental condition they were assigned to, used patterns to some extent in the interface designs.

In the previous sections we reviewed a number of pattern-like structures already being used in communication, comprehension, problem solving, design, and other areas. It seems that we as a community can learn from further investigation into the ways these pattern-like structures are being used in other areas. In addition to the many pattern-like structures described herein we see a similarity between patterns and other structures such as case studies, (like those described in *Harvard Business Review*, Thomas & Brubaker, 2001; Yin, 2003), concept maps (Leake, Maguitman, & Reichherzer, 2004), case-based reasoning (Aamodt & Plaza, 1994), and speech acts (Searle, 1969). Another interesting topic to explore is the difference between how we use and observe patterns and how we discuss and document them. These instances of pattern-like structures have helped clarify how patterns are being used in other areas but, we still need further exploration and empirical evidence within HCI. In the next Chapter we provide conclusions and plans for future research.

8. CONCLUSIONS AND FUTURE WORK

For more than two decades now the HCI literature has focused on the *promise of pattern languages* (Pemberton, 2000), or the possible benefits they may provide, but there has been little empirical work to support these claims (Dearden & Finlay, 2006). The results of this empirical study suggest that the value of pattern languages in HCI may not be in reuse, at the early stages of design, or in terms of the quality of the resulting design, or the time taken to design the interfaces, when designers are familiar with the domain. In the last Chapter we described how pattern languages are already being used today to support comprehension, communication, problem solving, and knowledge management. The results of this study and our brief review suggest that patterns are all around us and they have been around for much longer than Alexander's (1977; 1979) texts.

In this study, we examined the impact of an information retrieval pattern language on the design of information retrieval interfaces, in terms of the quality of the resulting designs. We observed that the patterns had no apparent affect on the quality of the resulting interfaces or the time taken to design the interfaces. We also observed that the participants had a similar view of the underlying relationships between the patterns in aIRPLane as suggested by the cluster analysis and MDS. We discovered a high occurrence of the patterns in aIRPLane in many popular web search engines and traditional information retrieval systems. We also discovered that all the participants, regardless of the condition they were assigned, used some of the patterns in aIRPLane in their interface designs. In addition, there was a significant positive correlation between the perceived quality ratings and the number of patterns present in the IR systems, search engines and the participant's interfaces, with higher quality ratings associated with a higher number of patterns.

Some of the limitations of this study include that the main method of evaluation was expert review and that the variables used to assess quality were highly correlated. The participants were also design students, not professional designers. In addition, the participants worked individually, not in pairs or groups therefore, there was no communication to analyze. Another limitation is that the participants only created an initial design or a first iteration. In light of these limitations we are interested in future work that attempts to not only address some of these limitations but also builds on what we have learned from this study.

We revisit our research questions, particularly research question 1, here in an effort to further guide our future work. Research question 1 addressed the impact of a pattern language on the design of information retrieval interfaces. After analyzing the results of this study this question still seems to be an important question and one that the community cares about but, it seems that we need to understand more about the context in which we are asking such a question when considering future work. We need to think about a number of things when considering the context in which we are asking such a question.

- Who is using the pattern language?
- At what point in the design process?
- What type of interfaces are they designing?
- What experience do they have using and designing systems in this domain?
- How complex is the task?
- At what level of fidelity and interactivity are they designing the interfaces?
- Quality in terms of what?

In terms of who is using the pattern language, we need to consider the participants' experience designing and evaluating systems. Clearly students, new practitioners, or very experienced designers have varying levels of experience which may impact how design techniques are used. We need to consider the stage in the design process at which we are asking the participants to design the interfaces. There are differences in designing the first iteration or a much later iteration (and if we are asking participants to design a much later iteration, was the previous iteration designed by someone else). We also need to consider the type of interfaces we are asking the participants to design. There are obviously differences in designing information retrieval interfaces, e-commerce interfaces, computer aided design interfaces, air traffic control interfaces, naval combat interfaces, or rescue robot interfaces.

In considering the experience the participants have using and designing interfaces we need to keep in mind whether they have a few months experience designing such systems, 20 years of experience, or if they simply just have some experience using similar systems. In the preface to *Designing Interfaces*, Jennifer Tidwell in describing the patterns in her book, points out “If you’ve done any web or UI design, or even thought much about it, you should say, ‘Oh, right, I know what that is’ to most of these patterns. But a few of them might be new to you, and some of the familiar ones may not be part of your usual design repertoire” (2006, p. xv). She is emphasizing that these patterns may not be new to people with some design experience. When choosing our populations for empirical studies we need to focus on understanding how much experience the participants have in designing or using similar systems.

The complexity of the design task must also be considered. There is clearly a difference between designing five information retrieval interfaces and 15 rescue robot interfaces. Here we must also consider the scope of the system and the functionality we are asking participants to design, in addition to the number of interfaces. The level of fidelity and interactivity required in a design should also be considered. Paper prototypes vary drastically from fully functioning prototypes. Our measurements of quality need to also be considered in light of all the other issues raised above. Quality may be assessed in a number of ways depending upon the complexity of the task, the interactivity of the prototypes, and in terms of the evaluation metrics and techniques chosen, for example expert review, heuristic evaluation, or usability testing.

For those reasons suggest above and more, in the future when formulating similar research questions, we, as a community, must carefully consider the context in which these questions are being asked. In the past few decades there have been significant contributions to the literature in terms of documenting patterns and pattern languages. In addition, there have been a few empirical contributions that have helped us further understand the value of pattern languages in HCI. We suggest that we, as a community, closely examine the current state of work in this area in an attempt to ensure our efforts are not misdirected. We are urging the community to focus on providing empirical support for the promises in the literature.

We suggest a research agenda that is an extension to the research agenda suggested by Dearden & Finlay (2006). Including evaluating the contribution that patterns can have on:

- Communication (between designers and users, between interaction designers and software designers, and so on)
- The design process (particularly later iterations of a design)
- The capturing and sharing of design knowledge and design rationale

We suggest examining the impact pattern languages have on communication between people with similar and diverse backgrounds. Here we may examine the impact of a pattern language on communication, between designers and users, and between different types of designers, in terms of problems identified, specifications identified, and changes to specifications. In addition, we suggest examining when these things take place within the design process to identify whether there may be a savings in time and effort.

We suggest a shift in thinking and a shift in focus, to one which explores how patterns impact the design process and not necessarily just the products of design. In examining the design process we suggest conducting longitudinal studies which examine the impact of a pattern language on different phases throughout the design process, for example throughout multiple design and evaluation iterations. More case studies or ethnographies documenting how designers actually use pattern languages in their daily work would help further our understanding of the value of patterns languages in HCI.

We also suggest that we, as a community, focus on ways of documenting and capturing patterns so that they may be used and shared. One of Alexander's goals in documenting a pattern language was to capture what he referred to as the *quality without a name*, the quality present in spaces that feel whole and alive. In architecture it is not easy to define and describe quality and in HCI it is not easy to define and describe usability but, in both domains patterns and pattern languages help us describe quality and usability in a way that others can understand. Being able to describe and articulate usability through the use of patterns is surely of great value. In trying to find ways to successfully document, share, and use pattern languages we suggest the community begins by focusing on groups of designers and/or users who have an explicit need for such a thing. We suggest starting by first understanding who the users will be, what there needs are, and how they may actually use a pattern language.

We do not have all the answers for how to go about doing this. We, as a community, need to work together to address these problems. We are simply advising the community to stop spending so much energy discussing the promises that pattern languages may provide and instead focus on providing empirical support for these claims. As a community we need to shift our focus to trying to better understand the value of pattern languages in HCI. In doing this we, as a community, will then begin to see the benefits from all the great efforts in this area.

REFERENCES

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Communications, 7*(1), 39-59.

Agarwal, M. & Cagan, J. (1998). A blend of different tastes: the language of coffeemakers. *Environment and Planning B: Planning and Design, 25*, 205-226.

Alexander, C. *Pattern language.com curriculum vitae.* http://www.patternlanguage.com/leveltwo/caframe.htm?/leveltwo/../bios/vitae.htm

Alexander, C. (1964). *Notes on the synthesis of form* Harvard University Press.

Alexander, C. (1979). *The timeless way of building*. New York: Oxford University Press.

Alexander, C. (1981). *The linz Cafe/Das linz cafe*. Oxford University Press: New York.

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

Alexander, C., Silverstein, M., Angel, S., & Abrams, D. (1975). *The oregon experiment*. New York: Oxford University Press.

Atwood, M. E., McCain, K. W., & Williams, J. C. (2002). How does the design community think about design? *DIS '02: Proceedings of the Conference on Designing Interactive Systems,* London, England. 125-132.

Barnum, C. M. (2002). *Usability testing and research*. New York: Longman.

Bartlett, F. C. (1995). *Remembering: A study in experimental and social psychology*. Cambridge: Cambridge University Press.

Bayle, E., Bellamy, R., Casaday, G., Erickson, T., Fincher, S., Grinter, B., et al. (1998). Putting it all together: Towards a pattern language for interaction design: A CHI 97 workshop. *SIGCHI Bulletin, 30*(1).

Belkin, N. J. (1980). Anomalous state of knowledge as a basis for information retrieval. *Canadian Journal of Information Science, 5,* 133-143.

Bevan, N. (1995). Measuring usability as quality of use. *Software Quality Journal,* (4), 155-160.

Beyer, H., & Holtzblatt, K. (1998). *Contextual design: Defining customer-centered systems*. San Francisco, California: Morgan Kaufmann.

Bobrow, D. G., & Whalen, J. (2002). Community knowledge sharing in practice: The eureka story. *Reflections, 4*(2).

Borchers, J. (2000). A pattern approach to interaction design. *Proceedings of the DIS 2000 International Conference on Designing Interactive Systems,* New York. 369-378.

Borchers, J. (2001). *A pattern approach to interaction design.* John Wiley & Sons.

Borchers, J. (2002). Teaching HCI design patterns: Experience from two university courses. Position Paper for "Patterns in Practice: A Workshop for UI Designers", *Workshop at CHI 2002 International Conference on Human Factors of Computing Systems*, Minneapolis, MI.

Borchers, J., Fincher, S., Griffiths, R., & Pemberton, L. Siemon, E. (2001). Usability pattern language: Creating a community. *AI & Society, 15*(4), 377-385.

Brooks, F. P. Jr. (1987). No silver bullet: Essence and accidents of software engineering. *IEEE Computer, 20*(4), 10-19.

Burns, E. (Oct 26, 2007). *Top 10 U.S. search providers, September 2007.* Retrieved Nov 11, 2007, from http://searchenginewatch.com/showPage.html?page=3627422

Carlile, P. R. (2004). Transferring, translating, and transforming: An integrative framework for managing knowledge across boundaries. *Organization Science*, *15*(5), 555-568.

Carroll, J. M. (2006). Introduction to this special issue on foundations of design in HCI. *Human-Computer Interaction, 21*(1), 1-3.

Casaday, G. (1997). Notes on a pattern language for interactive usability. *CHI 97Electronic Publications: Late-Breaking/Short Talks.*

Chase, S.C. (2002). A model for user interaction in grammar-based design systems. *Automation in Construction*, *11*(2), 161-172.

Chomsky, N. (1957). *Syntactic structures*. Hague, Netherlands: Mouton & Co. Publishers.

Chomsky, N. (1975). *The logical structure of linguistic theory*. New York, NY: Plenum Press.

Chiou, S.C. & Krishnamurti, R. (1995). The grammar of Taiwanese traditional vernacular dwellings. *Environment and Planning B: Planning and Design, 22*, 689-720.

Chung, E. S., Hong, J. I., Lin, J., Prabaker, M. K., Landay, J. A., & Liu, A. L. (2004). Development and evaluation of emerging design patterns for ubiquitous computing. *Proceedings of the Designing Interactive Systems Conference*, Cambridge, Massachusetts. 233-242.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences*. Hillsdale, N.J.: L. Erlbaum Associates.

Collection of pattern conferences, workshops and panels from 1997 to 2004. http://www.hcipatterns.org/events.html

Computer Professionals for Social Responsibility (CPSR). *A pattern language for living communication.*http://www.cpsr.org/prevsite/program/sphere/patterns/index.html

Cooke, N. J. (1994). Varieties of knowledge elicitation techniques. *International Journal of Human-Computers Studies, 41*(6), 801-849.

Coram, T., & Lee, J. *Experiences -- A pattern language for user interface design.* http://www.maplefish.com/todd/papers/experiences/Experiences.html

Cowley, L., & Wesson, J. (2005). An experiment to measure the usefulness of patterns in the interaction design process. *INTERACT 2005,* 1142-1145.

Crystal, D. (1997a). Grammar. *The Cambridge encyclopedia of language* (Second Ed.). University Press, Cambridge: Cambridge University Press, 88.

Crystal, D. (1997b). Language and other communication systems. *The Cambridge encyclopedia of language* (Second Ed.) New York, NY: Cambridge University Press, 400.

Dearden, A., Finlay, J., Allgar, L., & McManus, B. (2002a). Evaluating pattern languages in participatory design. *CHI '02: CHI '02 Extended Abstracts on Human Factors in Computing Systems,* Minneapolis, Minnesota, USA. 664-665.

Dearden, A., Finlay, J., Allgar, L., & McManus, B. (2002b). Using pattern languages in participatory design. *Participatory Design Conference (PDC) 2002,* Malmo, Sweden.

Dearden, A., & Finlay, J. (2006). Pattern languages in HCI: A critical review. *Human-Computer Interaction, 21,* 49-102.

Deng, J., Kemp, E., & Todd, E. G. (2005). Managing UI pattern collections. *CHINZ '05: Proceedings of the 6th ACM SIGCHI New Zealand Chapter's International Conference on Computer-Human Interaction,* Auckland, New Zealand. 31-38.

Dervin, B. (1983). An overview of sense-making research: Concepts, methods, and results to date. *Seattle: School of Communications, University of Washington,*

Dervin, B., & Nilan, B. (1986). Information needs and uses. *Annual Review of Information Science and Technology, 21,* 3-33.

Di Lucca, G. A., Fasolino, A. R., & Tramontana, P. (2005). Recovering interaction design patterns in web applications. 366-374.

Dictionary.com. Pattern.

Dovey, K. (1990). The pattern language and its enemies. *Design Studies, 11*(1), 3-9.

Downing F. & Flemming U. (1981). The bungalows of Buffalo. *Environment and Planning B: Planning and Design, 8*(3), 269-293.

E-commerce patterns for building B2C web sites using IBM websphere commerce suite V5.1 (2001). IBM.

Erickson, T. *The interaction design patterns page.* http://www.visi.com/%7Esnowfall/InteractionPatterns.html

Erickson, T. (2000). Lingua francas for design: Sacred places and pattern languages. *Proceedings of the ACM Conference on Designing Interactive Systems, DIS ’00,* Brooklyn, New York.

Fincher, S. (1999a). Analysis of design: An exploration of patterns and pattern languages for pedagogy. *Journal Comput. Math. Sci Teach., 18*(3), 331-348.

Fincher, S. (1999b). What is a pattern language? *Patterns Workshop at INTERACT, 1999.*

Fincher, S. (2002). Patterns for HCI and cognitive dimensions: Two halves of the same story? *Proceedings PPIG (Psychology of Programming Interest Group) Workshop 14.*

Fincher, S. (2003a). Perspectives on HCI patterns: Concepts and tools. *CHI 2003 Workshop Report.*

Fincher, S. (2003b). Perspectives on HCI patterns: Concepts and tools. CHI 2003 Workshop report. *Proceedings of CHI 2003,* http://www.cs.kent.ac.uk/people/staff/saf/patterns/chi2003/index.html.

Fincher, S., Finlay, J., Greene, S., Jones, L., Matchen, P., Thomas, J. C., et al. (2003). Perspectives on HCI patterns: Concepts and tools. *CHI '03: CHI '03 Extended Abstracts on Human Factors in Computing Systems,* Ft. Lauderdale, Florida, USA. 1044-1045.

Fincher, S., & Windsor, P. (2000). Why patterns are not enough: Some suggestions concerning an organizing principle for patterns of UI design. *CHI 200 Workshop on Pattern Languages for Interaction Design: Building Momentum.*

Finlay, J., Allgar, E., Dearden, A., & McManus, B. (2002). Pattern languages in participatory design. *People and Computers XVI—Memorable Yet Invisible, Proceedings of HCI2002,* London. 159-174.

Fischer, G., McCall, R., Ostwald, J., Reeves, B., & Shipman, F. (1994). Seeding, evolutionary growth and reseeding: Supporting the incremental development of design environments. *Proceedings SIGCHI 1994.*

Fowler, M. (1997). *Analysis patterns: Reusable object models*. Menlo Park, CA: Addison Wesley Longman, Inc.

Fromm, D., & Bosselmann, P. (1984). Mexicali revisited: Seven years later. *Places, 1*(4), 78-90.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns: Elements of reusable object-oriented software.* Addison-Wesley.

Garson, G.D., (n.d.). "Data Levels and Measurements", from *Statnotes: Topics in Multivariate Analysis.* Retrieved 05/06/2008 from http://www2.chass.ncsu.edu/garson/pa765/statnote.htm

Golden, E., John, B. E., & Bass, L. (2005a). Quality vs. quantity: Comparing evaluation methods in a usability-Focused Software architecture modification task. *Proceedings of the 4th International Symposium on Empirical Software Engineering,* Noosa Heads, Australia.

Golden, E., John, B. E., & Bass, L. (2005b). The value of a usability-supporting architectural pattern in software architecture design: A controlled experiment. *Proceedings of ICSE 2005 – 27th International Conference on Software Engineering,* St. Louis, MO, USA.

Gould, J. D., & Lewis, C. (1985). Designing for usability: Key principles and what designers think. *Communications of the ACM, 28*(3), 300-311.

Granlund, A., Lafreniere, D., & Carr, D. A. (2001). A pattern-supported approach to the user interface design process. *Proceedings of HCI International 2001 9th International Conference on Human-Computer Interaction,* New Orleans, USA.

Griffiths, R. *The Brighton usability pattern collection.* http://www.cmis.brighton.ac.uk/research/patterns/home.html

Griffiths, R., & Pemberton, L. (2005). *Don't write guidelines write patterns!* Retrieved August 4, 2006, from http://www.it.bton.ac.uk/staff/lp22/guidelinesdraft.html

Griffiths, R., Pemberton, L., Borchers, J., & Stork, A. (2000). Pattern languages for interaction design building momentum A CHI 2000 workshop. *Proceedings of CHI 2000.*

Guy, E. S. (2005). "…real, concrete, fact about what works…": Integrating evaluation and design through patterns. *Proceedings GROUP '05,* Sanibel Island, Florida, USA.

Hannah, S. (2005). Sorting out card sorting: Comparing methods for information architects, usability engineers and other practitioners. (Masters Capstone report, University of Minnesota, Crookston).

Henderson, A., & Kyng, M. (1991). There's no place like home: Continuing design in use. In J. Greenbaum, & M. Kyng (Eds.), *Design at work: Cooperative design of computer systems.* Hillsdale, N.J: Lawrence Erlbaum Associates.

Henninger, S., & Ashokkumar, P. (2005). An ontology based infrastructure for usability design patterns. *Workshop on Semantic Web Enabled Software Engineering (SWESE), 4Th International Semantic Web Conference ISCW 2005,* Galaway, Ireland. 41-55.

Henninger, S., & Corrêa, V. (2007). Software pattern communities: Current practices and challenges. *14th Conference on Pattern Languages of Programs (PLoP 07),* Monticello, IL.

Hewett, T., et al. (1996). *ACM SIGCHI curricula for human-computer interaction.* Retrieved August 25, 2005, from http://sigchi.org/cdg/cdg2.html#2_1

Hillside.net. *Hillside.net.*http://hillside.net/patterns/

Hoenig, J. M., & Heisey, D. M. (2001). The abuse of power: The pervasive fallacy of power calculations for data analysis. *The American Statistician*, *55*(1), 19-24.

Holzner, S. (2006). *Design patterns for dummies*. Hoboken, NJ: Wiley Publishing Inc.

IBM Patterns for e-business. *IBM patterns for e-business.* Retrieved October 16, 2006, from http://www-128.ibm.com/developerworks/patterns/

Jamieson, S. (2004). Likert scales: How to (ab)use them. *Medical Education, 38*(12), 1217-1218.

John, B. E., & Kieras, D. E. (1996). The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transaction on Computer-Human Interaction*, *3*(4), 320-351.

Jones, J. C. (1992). *Design methods* (2nd Ed.) Wiley.

Karat, J. (1997). User-centered software evaluation methodologies. In M. G. Helander, T. K. Landauer & P. V. Prabhu (Eds.), *Handbook of human-computer interaction.* (2nd Ed.) Englewood Cliffs, N.J.: Elsevier Science.

Kim, S. (1990). Interdisciplinary cooperation. In B. Laurel (Ed.) *The art of human-computer interface design.* Boston: Addison-Wesley. p. 31-44.

Kintsch, W. (1977). On comprehending stories. In M. A. Just, & P. A. Carpenter (Eds.), *Cognitive processes in comprehension.* Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers. p. 33-86.

Kintsch, W. (1998). *Comprehension: A paradigm for cognition*. Cambridge, UK: Cambridge University Press.

Kotze, P., Renaud, K. Koukouletsos, K., Khazaei, K., & Dearden, A. (2006). Patterns, anti-patterns and guidelines - effective aids to teaching HCI principles? *The First Joint BCS/IFIP WG13.1/ICS/EU CONVIVIO HCI Educators' Workshop*, Limerick, Ireland.

Kuhlthau, C. C. (1991). Inside the search process: Information seeking from the user's perspective. *Journal of the American Society for Information Science, 42*, 361-371.

Kyng, M. (1991). Designing for cooperation: Cooperating in design. *Communications of the ACM*, *34*(12), 65-73.

Lafreniere, D., & Hedenskog, A. (2001). Describing and using patterns for UI design. *Usability Professionals Association (UPA) Annual Conference, UPA Extended Proceedings,* Lake Las Vegas, Nevada.

Language. Retrieved 4/9, 2008, from http://www.merriam-webster.com/dictionary/language

Leacock, M., Malone, E., & Wheeler C. (2005). Implementing a pattern library in the real world: A yahoo! case study. *American Society for Information Science and Technology Information Architecture Summit,* Montréal, Québec, Canada.

Leake, D., Maguitman, A., & Reichherzer, T. (2004). Understanding knowledge models: Modeling assessment of concept importance in concept maps. *CogSci-04,* 785-800.

Lewis, C., & Rieman, J. (1994). *Task centered user interface design: A practical introduction*

Lewis, C., & Wharton, C. (1997). Cognitive walkthroughs. In M. G. Helander, T. K. Landauer & P. V. Prabhu (Eds.), *Handbook of human-computer interaction.* (2nd Ed.) Englewood Cliffs, N.J.: Elsevier Science.

Mahemoff, M. J., & Johnston, L. J. (1998a). Pattern languages for usability: An investigation of alternative approaches. *Asia-Pacific Conference on Human Computer Interaction (APCHI) 98 Proceedings,* 25-31.

Mahemoff, M. J., & Johnston, L. J. (1998b). Principles for a usability-oriented pattern language. *Proceedings of the 1998 Australasian Computer Human Interaction Conference,* Adelaide, SA, Australia. 132-139.

Mahemoff, M. J., & Johnston, L. J. (2001). Usability pattern languages: The “Language” aspect. *Interact '01,* Tokyo, Japan. 350-358.

Malhotra, Y. (2002). Why knowledge management systems fail? enablers and constraints of knowledge management in human enterprises. In C. W. Holsapple (Ed.), *Handbook of knowledge management*. Heidelberg: Springer-Verlag.

Mandler, J. M. (1984). *Stories, scripts, and scenes: Aspects of schema theory*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Marchionini, G. (1995). *Information seeking in electronic environments.* Cambridge Series on Human-Computer Interaction. Cambridge University Press.

Maurer, D., & Warfel, T. (2004). *Card sorting: A definitive guide.* Retrieved 4/8, 2008, from http://www.boxesandarrows.com/archives/card_sorting_a_definitive_guilde.php

Mayhew, D. (1999). *The usability engineering lifecycle: A practitioner's handbook for user interface design* Morgan Kaufmann Publishers.

McCain, K. W. (1990). Mapping authors in intellectual space: A technical overview. *Journal of the American Society for Information Science, 41*(6), 433-443.

Merriam-Webster Online Dictionary. *Pattern.* Retrieved October 6, 2006, from http://www.m-w.com/dictionary/pattern

Molich, R., & Nielsen, J. (1990). Improving a human-computer dialogue. *Communications of the ACM, 33*(3), 338-348.

Murphy, K. R., & Myors, B. (2004). *Statistical power analysis: A simple and general model for traditional and modern hypothesis tests*. Mahwah, N.J.: Lawrence Erlbaum.

Nielsen, J. *Ten usability heuristics.* Retrieved 11/15, 2006, from http://www.useit.com/papers/heuristic/heuristic_list.html

Nielsen, J. (1994a). Enhancing the explanatory power of usability heuristics. *CHI '94: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* Boston, Massachusetts, United States. 152-158.

Nielsen, J. (1994b). Heuristic evaluation. In J. Nielsen, & R. L. Mack (Eds.), *Usability inspection methods*. New York: John Wiley & Sons Ltd.

Nielsen, J. (2003). Usability 101: Introduction to usability. *Alertbox.* http://www.useit.com /alertbox/20030825.html

Nielsen, J. (2005). *Durability of usability guidelines.* Retrieved 11/15, 2006, from http://www.useit.com/alertbox/20050117.html

Nielsen, J., & Levy, J. (1994). Measuring usability: Preference vs. performance. *Communications of the ACM, 37*(4), 66-75.

Nielsen, J., & Sano, D. (1994). SunWeb: User interface design for sun microsystem's internal web. *Computer Networks and ISDN Systems, 28*(1-2), 179-188.

Norman, D. A. (1998). *The design of everyday things.* The MIT Press.

Norman, D. A. (2002). *The design of everyday things*. New York: Basic Books.

Norman, D. A., & Draper, S. W. (1986). *User centered system design: New perspectives on human-computer interaction*. Hillsdale, N.J.: L. Erlbaum Associates.

O'Keefe, D. (2007). Post hoc power, observed power, A priori power, retrospective power, prospective power, achieved power: Sorting out appropriate uses of statistical power analyses. *Communication Methods and Measures, 1*(4), 291-299.

Orr, J. E. (1996). *Talking about machines: An ethnography of a modern job.* Cornell University Press.

Pallant, J. (2004). *SPSS survival manual* (2nd Ed.) Open University Press.

*The pedagogical patterns project.*http://www.pedagogicalpatterns.org/

Pemberton, L. (2000). The promise of pattern languages for interaction design. *Human Factors Symposium,* Loughborough, UK.

Preece, J., Rogers, Y., & Sharp, H. (2002). *Interaction design: Beyond human-computer interaction* John Wiley & Sons, Inc.

Rheinfrank, J., & Evenson, S. (1996). Design languages. In T. Winograd, J. Bennett, De Young L. & B. Hartfield (Eds.), *Bringing design to software*. New York: ACM Press.

Rheinfrank, J., Hartman, W. R., & Wasserman, A. (1992). Design for usability: Crafting a strategy for the design of a new generation of Xerox copiers. In P. S. Adler, & T. Winograd (Eds.), *Usability: Turning technology into tools*. New York: Oxford University Press.

Rittel, H. W. J. (1984). Second-generation design methods. In N. Cross (Ed.), *Developments in design methodology*. Chichester: John Wiley & Sons. 317-327.

Rittel, H. W. J., & Webber, M. M. (1984). Planning problems are wicked problems. In N. Cross (Ed.), *Developments in design methodology*. Chichester: John Wiley & Sons Ltd.

Rosson, M. B., & Carroll, J. M. (2002). *Usability engineering: Scenario-based development of human computer interaction*. San Francisco, California: Morgan Kaufmann Publishers.

Rugg, G., & McGeorge, P. (1997). The sorting techniques: A tutorial paper on card sorts, picture sorts and item sorts. *Expert Systems, 14*(2), 80-93.

Salingaros, N. A. (2000). The structure of pattern languages. *Architectural Research Quarterly, 4*, 149-161.

Saponas, T. S., Prabaker, M. K., Abowd, G. D., & Landay, J. A. (2006). The impact of pre-patterns on the design of digital home applications. *Proceedings of Designing Interactive Systems 2006,* Penn State.

Schank, R. C. (1986). *Explanation patterns: Understanding mechanically and creatively*. Hillsdale, N.J.: L. Erlbaum Associates.

Schön, D. A. (1987). *Educating the reflective practitioner*. San Francisco, California: Jossey Bass Inc.

Schummer, T., Borchers, J., Thomas, J., & Zdun, U. (2004). Human-computer-human interaction patterns: Workshop on the human role in HCI patterns. *Proceedings of CHI 2004,* Vienna, Austria.

Searle, J. R. (1969). *Speech acts: An essay in the philosophy of language*. Cambridge: Cambridge U.P.

Seffah, A., & Javahery, H. (2002). On the usability of usability patterns. *Workshop Entitled Patterns in Practice, CHI, 2002.*

Shneiderman, B., & Plaisant, C. (2005). *Designing the user interface: Strategies for effective human-computer interaction* (Fourth Ed.). Boston, MA: Pearson Education, Inc. Addison Wesley.

Shrout, P. E., & Fleiss, J. L. (1979). Intraclass correlations: Uses in assessing rater reliability. *Psychological Bulletin*, *86*(2), 420-428.

Simon, H. A. (1984a). The structure of ill-structured problems. In N. Cross (Ed.), *Developments in design methodology*. Chichester: John Wiley & Sons Ltd.

Simon, H. A. (1984b). The structure of ill-structured problems. In N. Cross (Ed.), *Developments in design methodology*. Chichester: John Wiley & Sons. 145-166.

Simon, H. A. (1996). *The sciences of the artificial* (3rd Ed.) MIT Press.

Snyder, C. (2003). *Paper prototyping: The fast and easy way to design and refine user interfaces*. San Francisco, California: Morgan Kaufmann Publishers.

Star, S. L. (1990). The structure of ill-structured solutions: boundary objects and heterogeneous distributed problem solving. 37-54.

Stewart, T., & Travis, D. (2003). Guidelines, standards, and style guides. In J. A. Jacko, & A. Sears (Eds.), *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications* (pp. 991-1005). Mahwah, NJ: Lawrence Erlbaum Associates.

Tabachnick, B. G., & Fidell, L. S. (1996). *Using multivariate statistics* (3rd Ed.). Harper Collins Publishers.

Tannen, D. (1993). What's in a frame? surface evidence for underlying expectations. In D. Tannen (Ed.), *Framing in discourse* (pp. 14-56). New York, NY: Oxford University Press.

Tannen, D., & Wallat, C. (1993). Interactive frames and knowledge schemas in interaction. In D. Tannen (Ed.), *Framing in discourse*. New York, NY: Oxford University Press.

Thomas, R. M., & Brubaker, D. L. (2001). *Avoiding thesis and dissertation pitfalls: 61 cases of problems and solutions*. Westport, Conn.: Bergin & Garvey.

Tidwell, J. (2006). *Designing interfaces*. Sebastopol, CA: O'Reilly Media Inc.

Todd, E., Kemp, E., & Phillips, C. (2004). What makes a good user interface pattern language? *Conferences in Research and Practice in Information Technology, in CRPIT ’28: Proceedings of the Fifth Conference on Australasian User Interface,* Darlinghurst, Australia. 91–100.

Trask, R. L. (1999). *Key concepts in language and linguistics*. London: Routledge.

Tullis, T., & Wood, L. (2004). How many users are enough for a card-sorting study? *Proceedings UPA'2004,* Minneapolis, MN.

Upchurch, L., Rugg, G., & Kitchenham, B. (2001). Using card sorts to elicit webpage quality attributes. *IEEE Software, July\August.*

van Dijk, T. A. (1977). Semantic macro-structures and knowledge frames in discourse comprehension. In M. A. Just, & P. A. Carpenter (Eds.), *Cognitive processes in comprehension.* Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers. pp. 3-31.

van Duyne, D. K., Landay, J. A., & Hong, J. I. (2003). *The design of sites: Patterns, principles, and processes for crafting a customer-centered web experience*. Boston, MA: Addison-Wesley.

van Welie, M. *Patterns in interaction design.*http://www.welie.com/index.html

van Welie, M., & Klaasse, B. (2004). *Evaluating museum websites using design patterns.* Technical Report IR-IMSE-001

van Welie, M., Mullet, K., & McInerney, P. (2002). Patterns in practice: A workshop for UI designers. *Proceedings of CHI 2002,* Minneapolis, MN.

van Welie, M., & van der Veer, G.C. (2003). Pattern languages in interaction design: Structure and organization. *Interact 2003.*

Wania, C. E., Atwood, M. E., & McCain, K. W. (2006). How do design and evaluation interrelate in HCI research? *DIS '06: Proceedings of the 6th ACM Conference on Designing Interactive Systems,* University Park, PA, USA. 90-98.

WebPatterns.org. *WebPatterns.org.*http://webpatterns.org/index.html

Website Patterns. (2005). *Website patterns.* Retrieved 9/18, 2006, from http://iawiki.net/WebsitePatterns

Wellhausen, T. (2006). User interface design for searching: A pattern language. *Tenth European Conference on Pattern Languages of Programs,* Irsee, Germany.

Wesson, J., & Cowley, L. (2003). Designing with patterns: Possibilities and pitfalls. *Proceedings of the 2nd Workshop on Software and Usability Cross-Pollination: The Role of Usability Patterns, INTERACT 2003,* Zürich, Switzerland.

Winn, T., & Calder, P. (2002). Is this a pattern? *Software, IEEE, 19*(1), 59-66.

Yahoo design pattern library. http://developer.yahoo.com/ypatterns/

Yin, R. K. (2003). *Case study research: Design and methods*. Thousand Oaks, Calif.: Sage Publications.

Appendix A. A List of Pattern Collections and Pattern Languages

Brighton Usability Pattern Collection
http://www.cmis.brighton.ac.uk/research/patterns/home.html

Computing Professionals for Social Responsibility – A pattern language for living
Communication http://www.cpsr.org/prevsite/program/sphere/patterns/index.html

Group for User Interface Research - Projects - Ubicomp Design Patterns
http://guir.berkeley.edu/projects/patterns/

IBM’s patterns for e-commerce
http://www-128.ibm.com/developerworks/patterns/

The Pedagogical Patterns Project
http://www.pedagogicalpatterns.org/

PoInter – Patterns of Interaction
http://www.comp.lancs.ac.uk/computing/research/cseg/projects/pointer/pointer.html

Portland Pattern Repository
http://c2.com/ppr/

Rising’s - Patterns Almanac
http://www.smallmemory.com/almanac/

Tidwell’s pattern collection
http://designinginterfaces.com/

van Duyne, Landay, and Hong - The Design of Sites
http://www.designofsites.com/

van Welie’s patterns in interaction design collection
http://www.welie.com/

Yahoo’s Design Pattern Library
http://developer.yahoo.com/ypatterns/

Appendix B. Pre Task Questionnaire

Thank you for your participation. Please answer the following questions by placing a checkmark in the appropriate box.

1. What is your age?
[   ] 18 – 25
[   ] 26 – 35
[   ] 36 – 45
[   ] 46 – 55
[   ] 55 and over

2. What degree are you currently pursuing?
[   ] Bachelor’s
[   ] Master’s
[   ] Doctoral

3. What is your current major?
____

4. Which Human-Computer Interaction courses have you taken at Drexel University?
[   ] INFO 110 - Human-Computer Interaction I
[   ] INFO 310 - Human-Computer Interaction II
[   ] INFO 608 - Human-Computer Interaction
[   ] INFO 610 - Analysis of Interactive Systems
[   ] INFO 611 - Design of Interactive Systems
[   ] Other:
please specify ____

5. How many Human-Computer Interaction courses have you taken at another university?
[   ] 0 courses
[   ] 1 – 2 courses
[   ] 3 – 5 courses
[   ] 6 – 8 courses
[   ] 9 or more courses

6. How many months/years job experience do you have designing/evaluating systems?
[   ] 0 – 6 months
[   ] 7 – 12 months
[   ] 1 – 2 years
[   ] 3 – 5 years
[   ] 5 or more years

Thanks again for your participation.

Appendix C. Design Task

You have been hired to design an information retrieval system. The information retrieval system must allow users to perform general and advanced searches. The information retrieval system must present the results of a search to the user and the system must also provide help for the user.

Please read the scenario below and design interfaces to illustrate how a user would interact with the system. Please make sketches of the interfaces and be specific about how the user is able to interact with the system in a given interface.

Scenario
Chris is a new student at The College of Information Science and Technology at Drexel University. Chris has been given his first assignment for his first course. The assignment is to find articles about Human Computer Interaction methods and techniques written by Pat Smith.

Chris goes to the Drexel Library and asks the librarian where he could find this type of information. The librarian directs Chris to the new College of IST library information retrieval system. The system allows users to search all the Library's electronic materials appropriate for the College of IST through one interface. Chris sits down and begins using the system.

- Chris searches for Pat Smith.
- Chris looks through the results and sees that the information does not seem relevant.
- Chris decides to try looking in the Help.
- After reading some of the help Chris performs an advanced search.
- Chris begins reading through the results and thinks that he may have found what he was looking for. Chris thanks the librarian and leaves the library.

Please sketch an interface for each step described in the above scenario.

Please be specific about how a user may interact with the system. Evaluators will be rating the functionality of the resulting interface designs. They should be able to understand how to interact with the system.

Appendix D. Patterns in HCI Overview

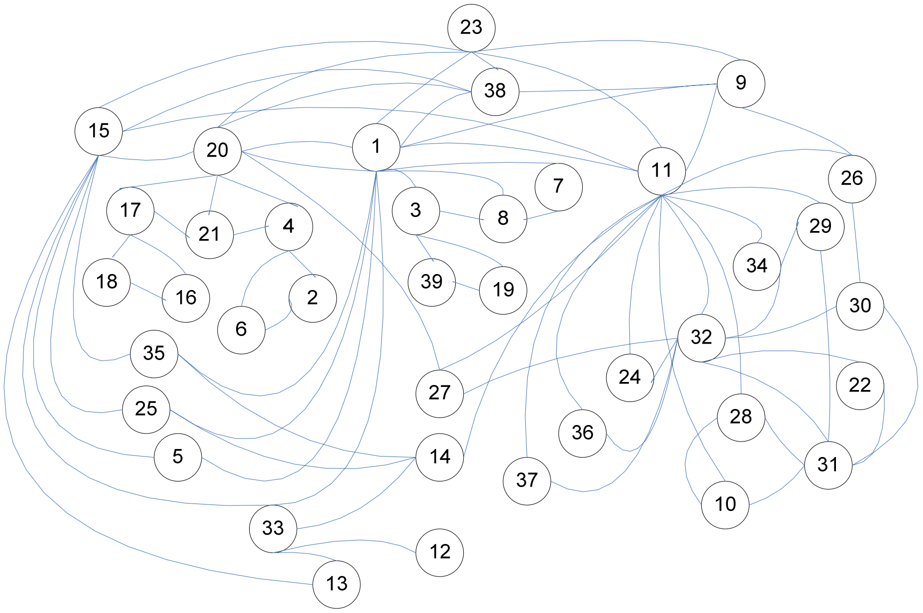
Overview of Patterns in HCI

Patterns can be used to design systems in HCI. A pattern is a solution to a design problem which occurs over an over again in an information system, website, or other system interface (for example, a cell phone). The primary goal of a pattern is to document a solution to a design problem that is common, difficult, and frequently encountered.

Each interaction design pattern contains the following elements:

- a name
- a picture
- what (describes what the pattern is)
- use when (describes when the pattern should be applied)
- why (describes how the solution solves the problem in a particular context)
- how (describes how the pattern may be applied)
- examples
- how this pattern is related to others (describes how the pattern is related to other patterns in a pattern language)

Appendix E. An Information Retrieval Pattern Language



1 Adv search
2 Adv search ex
3 Adv search fields
4 Adv search help
5 Adv search link
6 Adv search tips
7 Adv with field codes
8 Adv without field codes
9 Browse

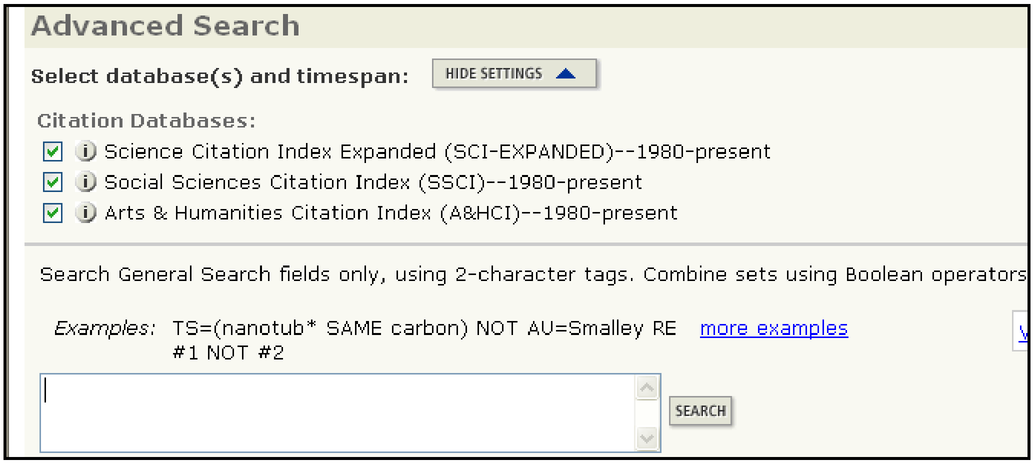
10 Display format
11 Display results
12 Execute on Enter or Search
13 Execute on enter
14 Executed query displayed
15 Gen search
16 Gen search example
17 Gen search help
18 Gen search tips
19 Grouping Adv search fields

20 Help
21 Help window
22 Indicate search terms
23 IR system
24 Mark results
25 Modify search
26 Navigation of results
27 No results
28 Num results per page
29 Num of retrieved results

30 Position in results
31 Result record
32 Result set
33 Search box and button
34 Search history
35 Search within results
36 Sort results by
37 Sorting order
38 Tab navigation
39 Tab through fields

Figure E – 1. aIRPLane Network Diagram

Advanced Search Example



Advanced Search

Select database(s) and timespan:

Citation Databases:

☑ Science Citation Index Expanded (SCI-EXPANDED)--1980-present
☑ Social Sciences Citation Index (SSCI)--1980-present
☑ Arts & Humanities Citation Index (A&HCI)--1980-present

Search General Search fields only, using 2-character tags. Combine sets using Boolean operators

Examples: TS=(nanotub* SAME carbon) NOT AU=Smalley RE
#1 NOT #2

From http://apps.isiknowledge.com/

What:

Provide examples of advanced searches that use common fields to quickly illustrate what can be done with an advanced search.

Use When:

When there is an advanced search in the system suitable examples should be provided. When trying to illustrate what types of advanced searches can be performed this is appropriate.

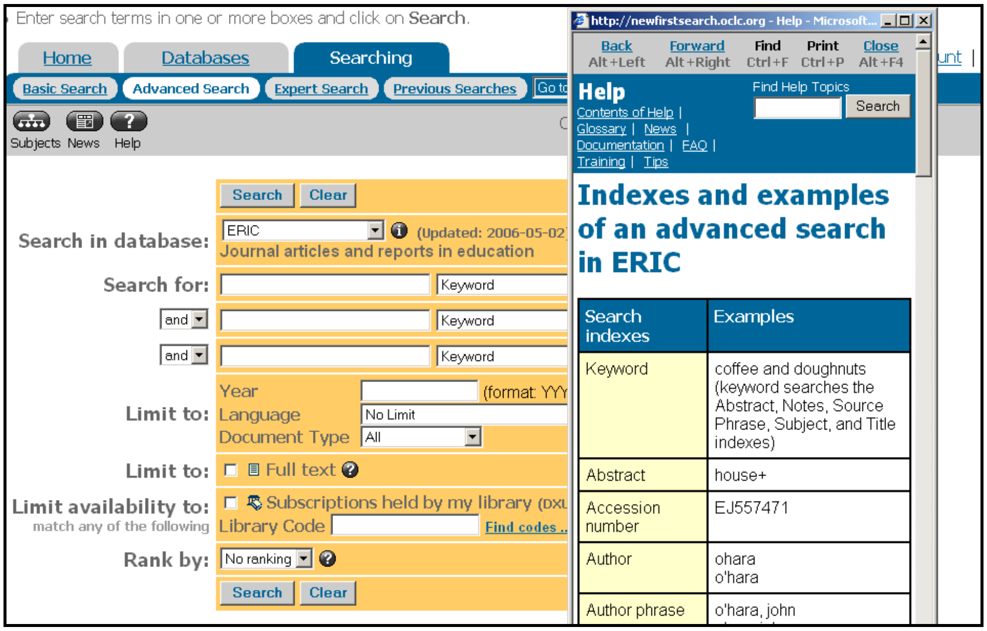
Why:

We would like to illustrate what a possible advanced search may look like in order to illustrate the types of advanced searches that can be performed using the system. We would like to allow users that are not familiar with advanced searches to take advantage of the power of advanced searches without having to spend a long time learning how to construct an advanced search. We do not want to only provide a tutorial and a help function because these may take up too much of the users' time, but we would like to provide some quick information about how to construct an advanced search.

How:

Provide an example of an advanced search directly in the search interface next to the appropriate advanced search fields and/or provide an example that can be accessed through the advanced search help function or separately through the help function. Provide a link on the advanced search page that takes the users directly to the example that illustrates how to use the most common features of an advanced search. This example should appear in a new window so that the user can read the help and look at the interface at the same time.

Examples:



Enter search terms in one or more boxes and click on Search.

Search in database: ERIC (Updated: 2006-05-02) Journal articles and reports in education

Search for: Keyword

and Keyword

and Keyword

Limit to: Year (format: YYY; Language No Limit; Document Type All

Limit to: □ Full text

Limit availability to: □ Subscriptions held by my library (DXU
match any of the following: Library Code Find codes ...

Rank by: No ranking

Indexes and examples of an advanced search in ERIC

Search indexes	Examples
Keyword	coffee and doughnuts (keyword searches the Abstract, Notes, Source Phrase, Subject, and Title indexes)
Abstract	house+
Accession number	EJ557471
Author	ohara o'hara
Author phrase	o'hara, john

From http://www.eric.ed.gov/ERICWebPortal/Home.portal

Examples

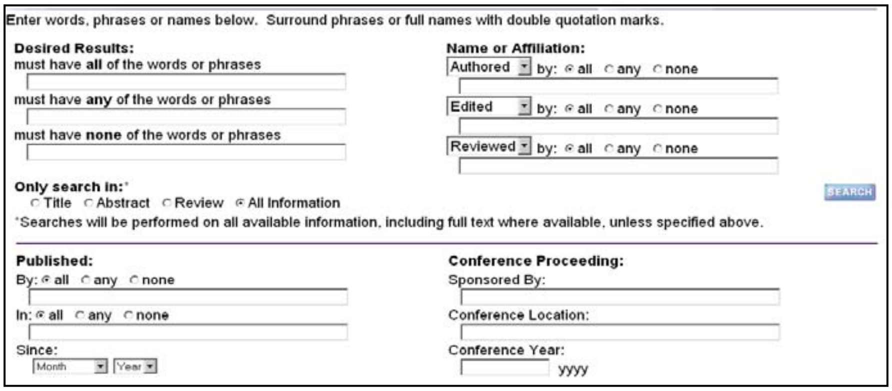
- TS=myoblast* AND AU=Zimmerman R*
- SO=(Journal of Ecology OR Ecology Letters_ AND TI=birch
- #2 AND #4
- (#1 OR #2) NOT #3

From http://apps.isiknowledge.com/

How this pattern is related to others:

This pattern is a part of **Advanced Search Help.** This pattern is related to **Advanced Search Tips.**

Advanced Search Fields



Enter words, phrases or names below. Surround phrases or full names with double quotation marks.

Desired Results:

must have **all** of the words or phrases

must have **any** of the words or phrases

must have **none** of the words or phrases

Name or Affiliation:

Authored by: ● all ○ any ○ none

Edited by: ● all ○ any ○ none

Reviewed by: ● all ○ any ○ none

Only search in:*

○ Title ○ Abstract ○ Review ● All Information

*Searches will be performed on all available information, including full text where available, unless specified above.

Published:

By: ● all ○ any ○ none

In: ● all ○ any ○ none

Since: Month Year

Conference Proceeding:

Sponsored By:

Conference Location:

Conference Year: yyyy

From http://portal.acm.org/dl.cfm

What:

Provide multiple advanced search fields (for example author, title, publication title, abstract, index terms, etc). Provide advanced search fields that allow the user to search for documents by entering information into one field or many different fields. Allow users to provide multiple restrictions by using an AND, OR, NOT functions. Allow users to restrict fields, for example publication dates and document types.

Use When:

When the system contains an advanced search provide the users with the option of expressing and entering their queries in many different ways (fields and formats).

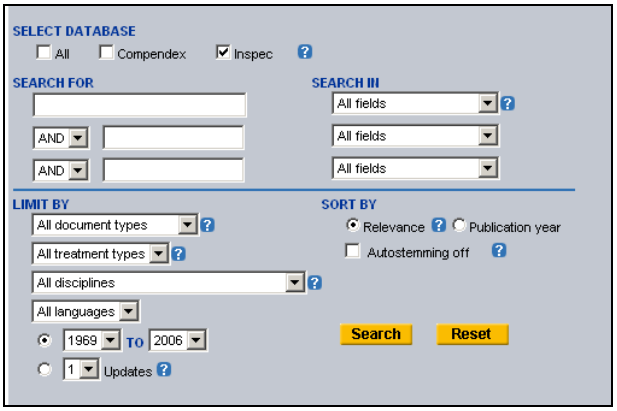
Why:

Users should be able to express their queries and narrow their search in a number of ways not just one. Users should not be required to search by all possible fields rather users should be allowed to limit the number of fields they search by. Providing multiple search fields allows for flexible searching (for example: by author and title, by author and year, by year and index terms, etc).

How:

Provide multiple advanced search fields that allow users to enter search terms in any or all of the fields. Provide fields for author, title, publication, keyword, etc. Allow users to provide multiple restrictions by using an AND, OR, NOT functions. For example allow users to search for “Joe” in the author field AND “soccer” in the keyword field. Also allow users to restrict things like publication dates (for example only articles published between 1990 and 2000) and document type (for example peer-reviewed articles, conference proceedings, etc). Provide descriptions for all the advanced search fields. Group advanced search fields visually so that users can see how the fields are related to one another. Provide advanced search options that are easy for the users to fill in. Provide descriptions, labels, or prompts for all advanced search fields in order to aid the user in expressing their query and knowing where to enter what information.

Examples:



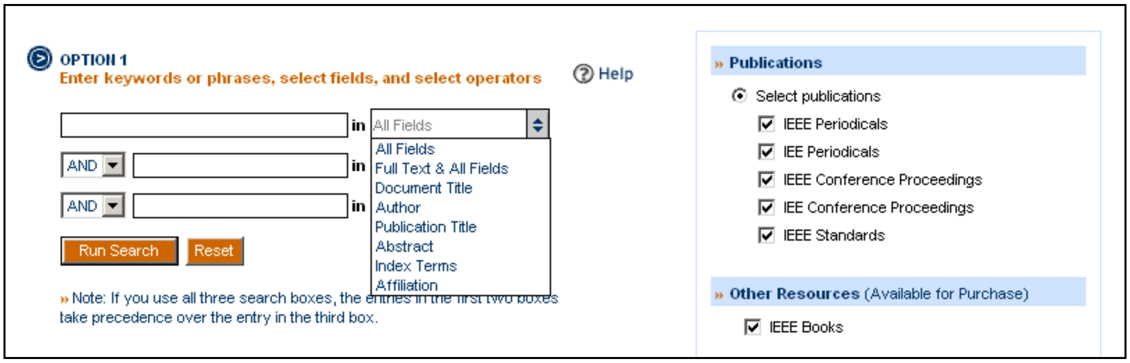
SELECT DATABASE
☐ All ☐ Compendex ☑ Inspec

SEARCH FOR | SEARCH IN
AND | All fields
AND | All fields
All fields

LIMIT BY
All document types
All treatment types
All disciplines
All languages
● 1969 TO 2006
○ 1 Updates

SORT BY
● Relevance ○ Publication year
☐ Autostemming off

From http://www.engineeringvillage2.org/



OPTION 1
Enter keywords or phrases, select fields, and select operators

in All Fields
AND in
AND in

All Fields
Full Text & All Fields
Document Title
Author
Publication Title
Abstract
Index Terms
Affiliation

» Note: If you use all three search boxes, the entries in the first two boxes take precedence over the entry in the third box.

» Publications
● Select publications
☑ IEEE Periodicals
☑ IEE Periodicals
☑ IEEE Conference Proceedings
☑ IEE Conference Proceedings
☑ IEEE Standards

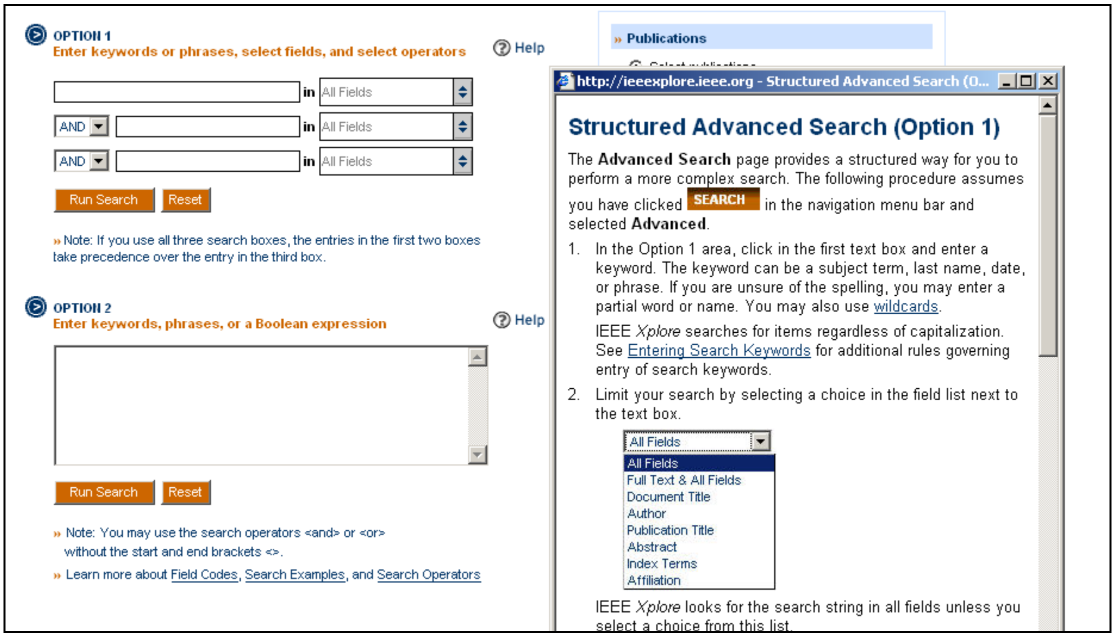
» Other Resources (Available for Purchase)
☑ IEEE Books

From http://ieeexplore.ieee.org/Xplore/dynhome.jsp

How this pattern is related to others:

This pattern is a part of **Advanced search**. This pattern is related to advanced search without tags. This pattern contains **Tab through fields** option and **Grouping advanced search fields**.

Advanced Search Help



» Note: If you use all three search boxes, the entries in the first two boxes take precedence over the entry in the third box.

» Note: You may use the search operators <and> or <or> without the start and end brackets <>.

» Learn more about Field Codes, Search Examples, and Search Operators

Structured Advanced Search (Option 1)

The **Advanced Search** page provides a structured way for you to perform a more complex search. The following procedure assumes you have clicked SEARCH in the navigation menu bar and selected **Advanced**.

1. In the Option 1 area, click in the first text box and enter a keyword. The keyword can be a subject term, last name, date, or phrase. If you are unsure of the spelling, you may enter a partial word or name. You may also use wildcards.

   IEEE *Xplore* searches for items regardless of capitalization. See Entering Search Keywords for additional rules governing entry of search keywords.

2. Limit your search by selecting a choice in the field list next to the text box.

   IEEE *Xplore* looks for the search string in all fields unless you select a choice from this list.

From http://ieeexplore.ieee.org/Xplore/dynhome.jsp

What:

Provide help specifically for an advanced search that explains how in detail how to construct an advanced search. The advanced search help should contain information that aids the users in constructing and executing advanced search queries.

Use When:

When the system has an advanced search it is necessary to provide help. When there will be users with varying levels of search experience using the system it is appropriate to provide information that will allow even novice searchers to take advantage of the advanced search features within the system.

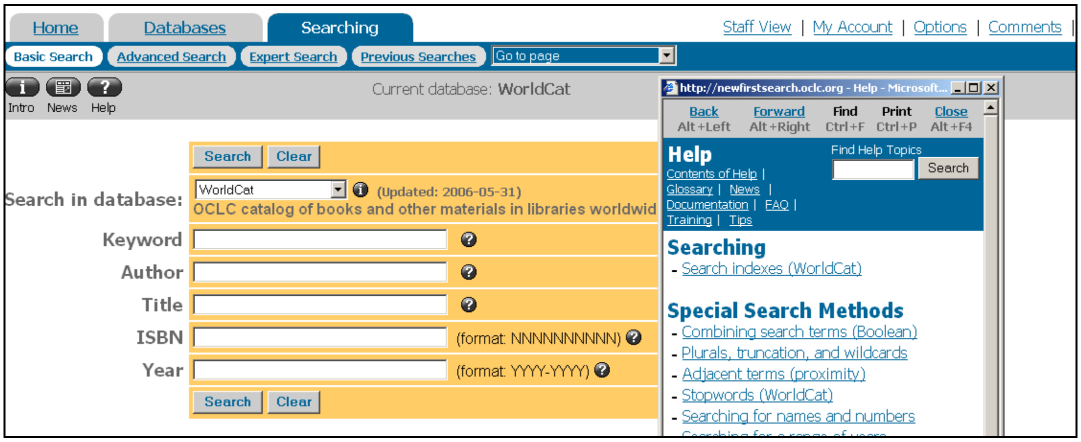
Why:

We would like to allow all users including users that are not familiar with constructing advanced searches to take advantage of the power of an advanced search. It is necessary to provide information that may assist users in constructing and executing advanced searches.

How:

Provide a help function that users can access while on the advanced search interface and from any other part of the system. The help function should provide a tutorial that explains how to execute an advanced search and explains how the advanced search works. This help function should also contain examples of advanced searches and search tips. This help function should appear in a new window so that the user can read the help and look at the interface at the same time.

Examples:



Current database: WorldCat

WorldCat (Updated: 2006-05-31)
OCLC catalog of books and other materials in libraries worldwid

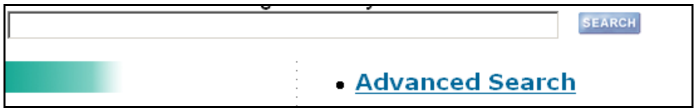
ISBN (format: NNNNNNNNNN)
Year (format: YYYY-YYYY)

From http://newfirstsearch.oclc.org/

How this pattern is related to others:

This pattern is a part of **Help** and **Advanced Search**. This pattern contains **Advanced search examples** and **Advanced Search Tips**. This pattern is related to **Help Window**.

Advanced Search Link



From http://portal.acm.org/dl.cfm

What:

Provide users with a link to the advanced search interface from the general search interface.

Use When:

When an information retrieval system has both a general search and an advanced search there should be a link to the advanced search interface from the general search interface.

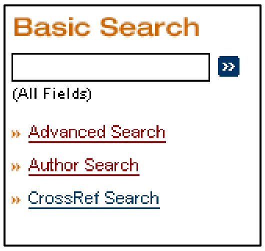
Why:

The general search is usually the default search in most systems therefore the general search interface is the first interface the user is presented. It is necessary to provide users with an easy way to navigate from the general search interface to the advanced search interface.

How:

Provide an advanced search link in the general search interface. Be sure that the advanced search link appears close to the search box and button in the general search interface.

Examples:

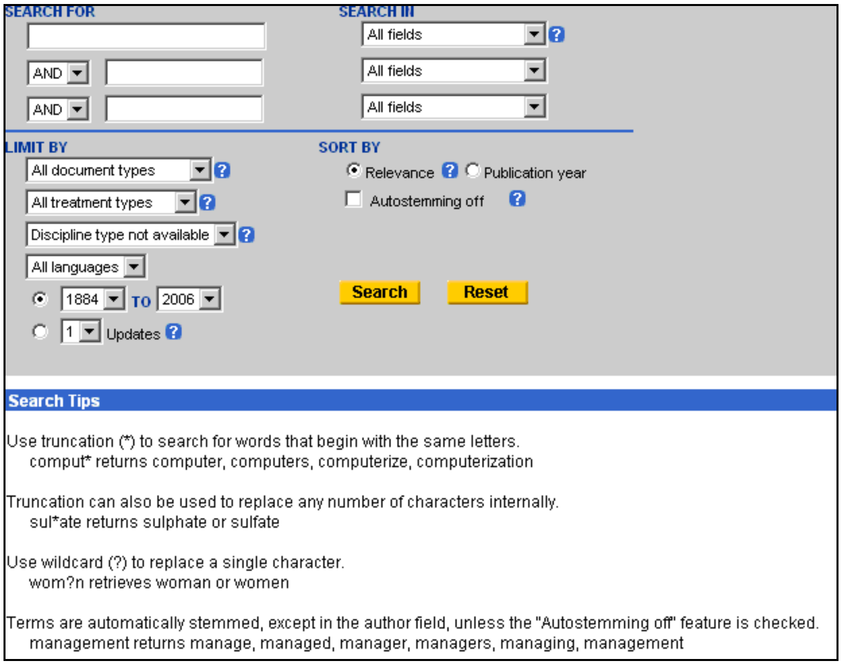


From http://ieeexplore.ieee.org/Xplore/dynhome.jsp

How this pattern is related to others:

This pattern is part of **General Search.** This pattern is related to **Advanced Search**.

Advanced Search Tips



Use truncation (*) to search for words that begin with the same letters.
comput* returns computer, computers, computerize, computerization

Truncation can also be used to replace any number of characters internally.
sul*ate returns sulphate or sulfate

Use wildcard (?) to replace a single character.
wom?n retrieves woman or women

Terms are automatically stemmed, except in the author field, unless the "Autostemming off" feature is checked.
management returns manage, managed, manager, managers, managing, management

From http://www.engineeringvillage2.org/

What:

Provide tips for advanced searching in the advanced search help. In addition to the tips in the advanced search help quick tips can also be provided directly on the advanced search interface.

Use When:

When trying to provide quick help in the advanced search interface tips are appropriate. When a system contains an advanced search there should also be tips included either in the interface or in the help or both. When there may be users with varying levels of experience and little time to learn how to use the system tips are appropriate.

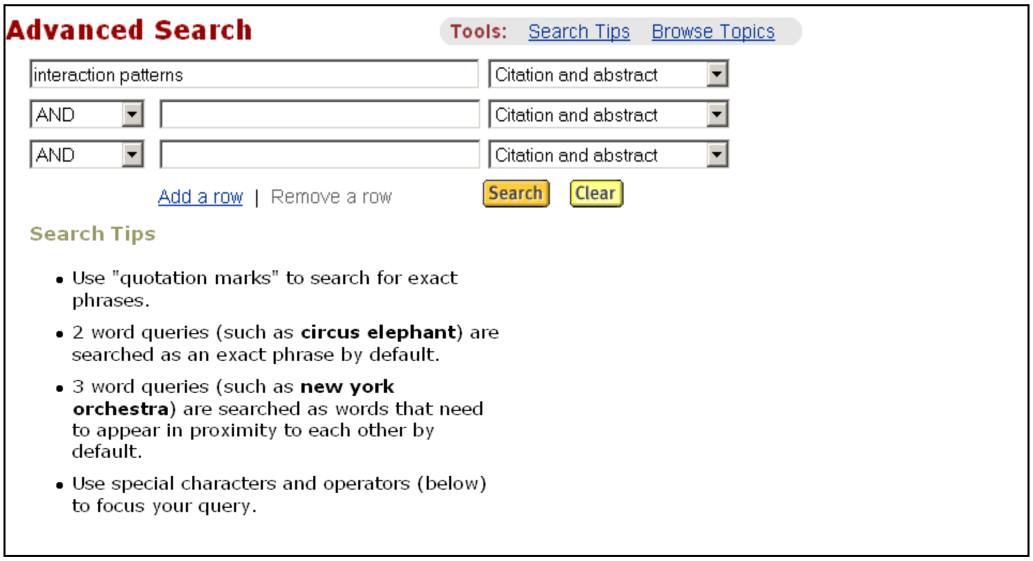
Why:

In order to help users, that may not be familiar with the system, get started using the system provide search tips in the advanced search interface. For those users that do not have time to read the help documentation this provides quick help directly in the interface that can help a user get started without spending a lot of time reading.

How:

Provide a link to advanced search tips in the advanced search interface and/or provide a few tips near the search fields.

Examples:


Advanced Search

Tools: Search Tips Browse Topics

interaction patterns | Citation and abstract

AND | | Citation and abstract

AND | | Citation and abstract

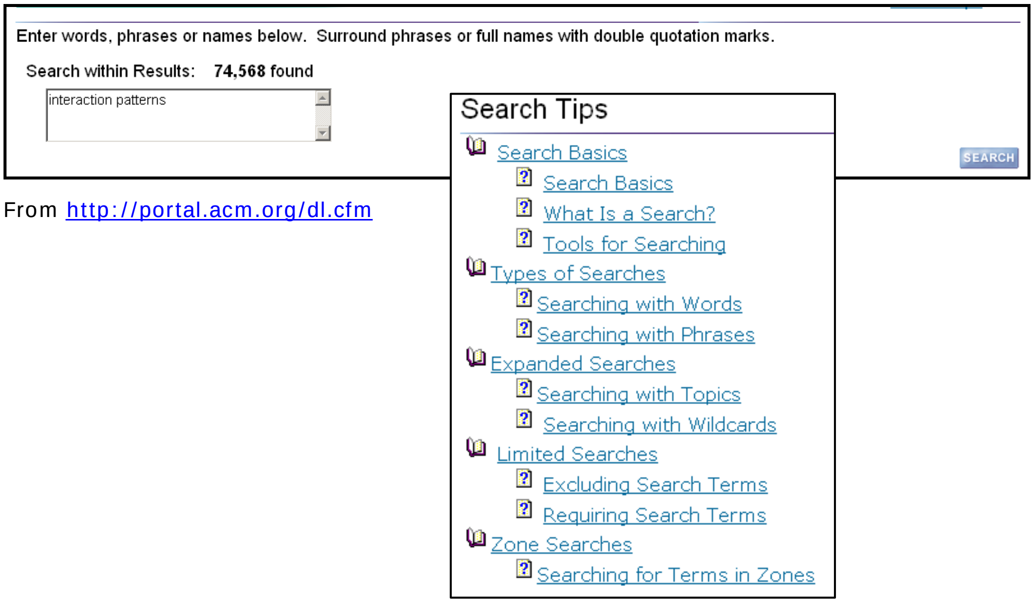
Add a row | Remove a row

Search Clear

Search Tips

- Use "quotation marks" to search for exact phrases.
- 2 word queries (such as **circus elephant**) are searched as an exact phrase by default.
- 3 word queries (such as **new york orchestra**) are searched as words that need to appear in proximity to each other by default.
- Use special characters and operators (below) to focus your query.

From http://proquest.com/


Enter words, phrases or names below. Surround phrases or full names with double quotation marks.

Search within Results: **74,568 found**

interaction patterns

SEARCH

Search Tips

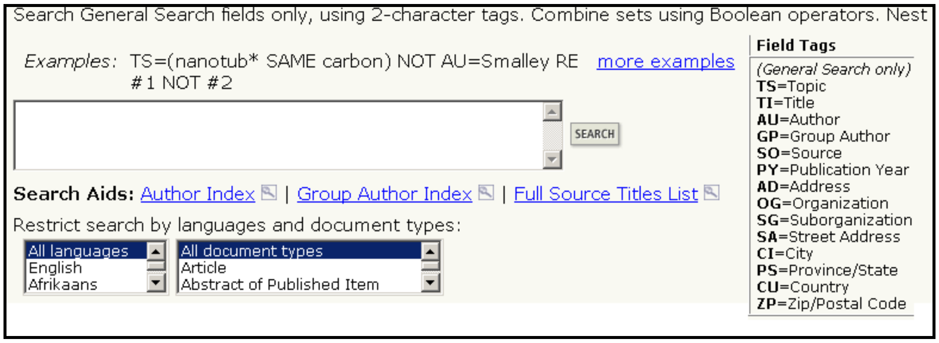
- Search Basics
  - Search Basics
  - What Is a Search?
  - Tools for Searching
- Types of Searches
  - Searching with Words
  - Searching with Phrases
- Expanded Searches
  - Searching with Topics
  - Searching with Wildcards
- Limited Searches
  - Excluding Search Terms
  - Requiring Search Terms
- Zone Searches
  - Searching for Terms in Zones

From http://portal.acm.org/dl.cfm

How this pattern is related to others:

This pattern is a part of **Advanced Search** and **Advanced Search Help**.

Advanced Search with Field codes



Search General Search fields only, using 2-character tags. Combine sets using Boolean operators. Nest

Examples: TS=(nanotub* SAME carbon) NOT AU=Smalley RE more examples
#1 NOT #2

Search Aids: Author Index | Group Author Index | Full Source Titles List

Restrict search by languages and document types:

Field Tags
(General Search only)
TS=Topic
TI=Title
AU=Author
GP=Group Author
SO=Source
PY=Publication Year
AD=Address
OG=Organization
SG=Suborganization
SA=Street Address
CI=City
PS=Province/State
CU=Country
ZP=Zip/Postal Code

From http://apps.isiknowledge.com/

What:

Provide users the option of constructing and executing advanced searches using field codes.

Use When:

When designing an IR system that can process advanced queries include an advanced search option that allows the use of field codes. When there is a possibility that advanced and expert users will use the system it is important to provide an advanced search with field codes.

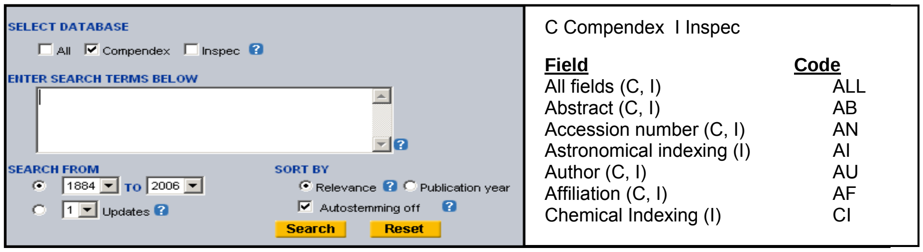
Why:

Providing an advanced search with field codes gives more advanced users the opportunity to quickly enter search queries and quickly utilize the power of an advanced search.

How:

Provide a large search box were users can enter multiple field codes and values. Also provide a summary of the field codes and their abbreviations near the search box.

Example:

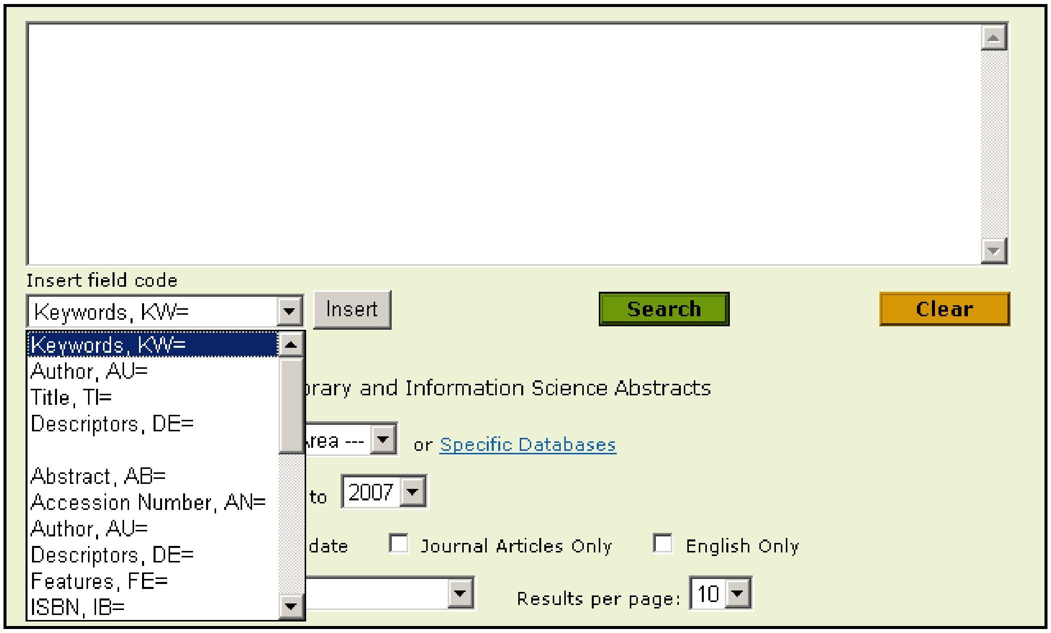


C Compendex I Inspec

Field	**Code**
All fields (C, I)	ALL
Abstract (C, I)	AB
Accession number (C, I)	AN
Astronomical indexing (I)	AI
Author (C, I)	AU
Affiliation (C, I)	AF
Chemical Indexing (I)	CI

From http://www.engineeringvillage2.org/

Example:



Insert field code: Keywords, KW=

to 2007

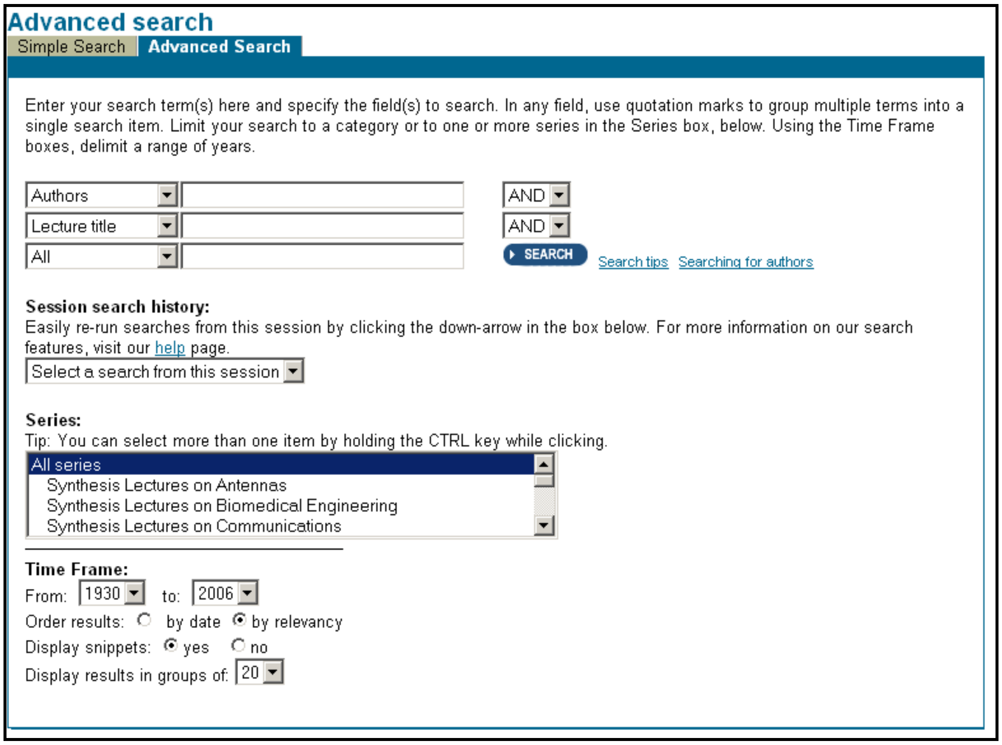
Results per page: 10

From http://www-md3.csa.com/csaillumina/login.php

How this pattern is related to others:

This pattern is a part of **advanced search**. This pattern is related to **advanced search without field codes**.

Advanced Search without Field codes



Enter your search term(s) here and specify the field(s) to search. In any field, use quotation marks to group multiple terms into a single search item. Limit your search to a category or to one or more series in the Series box, below. Using the Time Frame boxes, delimit a range of years.

Authors AND
Lecture title AND
All

Session search history:
Easily re-run searches from this session by clicking the down-arrow in the box below. For more information on our search features, visit our help page.
Select a search from this session

Series:
Tip: You can select more than one item by holding the CTRL key while clicking.
All series
Synthesis Lectures on Antennas
Synthesis Lectures on Biomedical Engineering
Synthesis Lectures on Communications

Time Frame:
From: 1930 to: 2006
Order results: ○ by date ● by relevancy
Display snippets: ● yes ○ no
Display results in groups of: 20

From http://www.morganclaypool.com/search/advanced

What:

Provide users with the option of using an advanced search that does not require the use of field codes. Present prompts for the necessary fields so the users do not need to remember field codes.

Use When:

When designing an IR system that can process advanced search queries include an advanced search option that does not require the use of field codes. When there is the possibility that less experiences users will use the system it is important to provide the functionality of an advanced search without making it too difficult to construct an advanced search query.

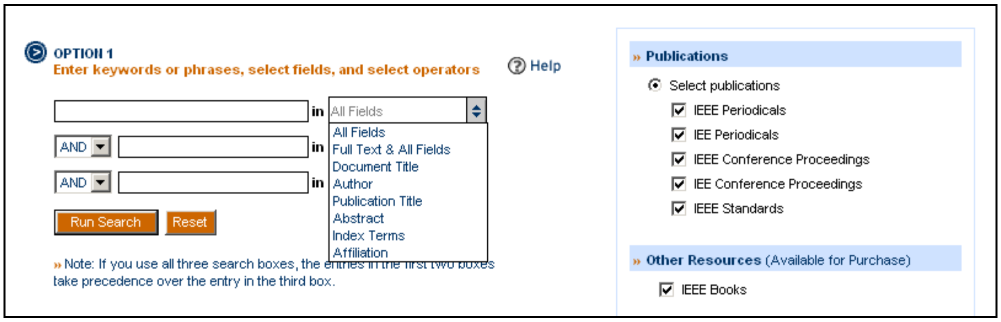
Why:

Users with little or no experience should be able to construct an advanced search without having to spend time learning field codes.

How:

Provide an advanced search with fields that prompt the user for the relevant information. If there are only a few possible advanced search fields use pull down menus for the selection of author, title, etc. instead of requiring the use of field codes. If there are many advanced search options provide a field that prompts the user for every possible advanced search field.

Examples:



OPTION 1
Enter keywords or phrases, select fields, and select operators

Help

in All Fields

AND in

AND in

All Fields
Full Text & All Fields
Document Title
Author
Publication Title
Abstract
Index Terms
Affiliation

Run Search | Reset

» Note: If you use all three search boxes, the entries in the first two boxes take precedence over the entry in the third box.

» Publications

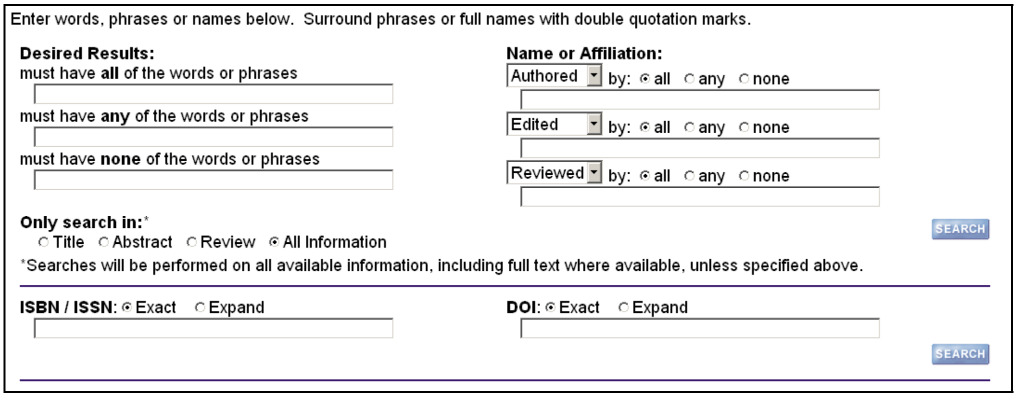
● Select publications

☑ IEEE Periodicals
☑ IEE Periodicals
☑ IEEE Conference Proceedings
☑ IEE Conference Proceedings
☑ IEEE Standards

» Other Resources (Available for Purchase)

☑ IEEE Books

From http://ieeexplore.ieee.org/Xplore/dynhome.jsp



Enter words, phrases or names below. Surround phrases or full names with double quotation marks.

Desired Results:
must have **all** of the words or phrases
must have **any** of the words or phrases
must have **none** of the words or phrases

Name or Affiliation:
Authored by: ● all ○ any ○ none
Edited by: ● all ○ any ○ none
Reviewed by: ● all ○ any ○ none

Only search in:*
○ Title ○ Abstract ○ Review ● All Information
*Searches will be performed on all available information, including full text where available, unless specified above.

SEARCH

ISBN / ISSN: ● Exact ○ Expand

DOI: ● Exact ○ Expand

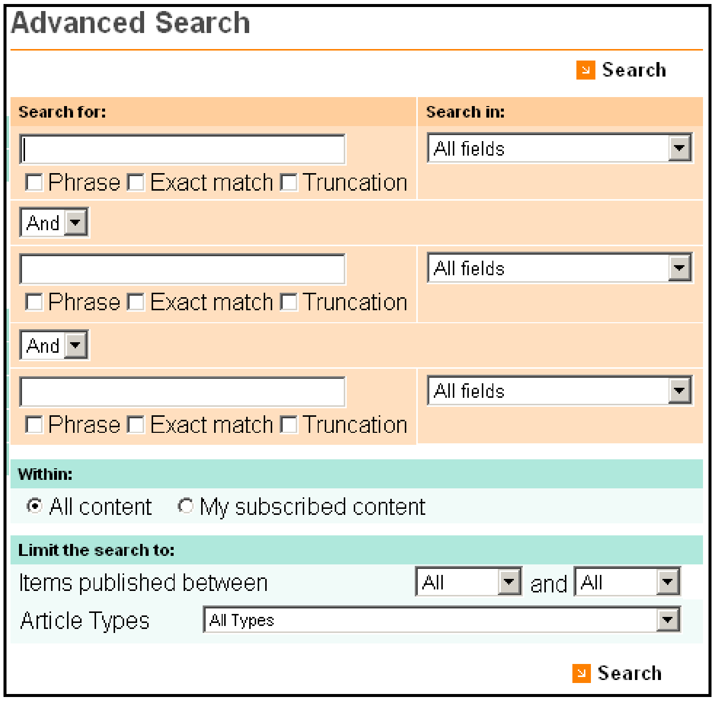
SEARCH

From http://portal.acm.org/dl.cfm

How this pattern is related to others:

This pattern is a part of **advanced search**. This pattern is related to advanced search fields. This pattern is related to **advanced search with codes**.

Advanced Search



Advanced Search

Search for: | Search in:

□ Phrase □ Exact match □ Truncation — All fields

And

□ Phrase □ Exact match □ Truncation — All fields

And

□ Phrase □ Exact match □ Truncation — All fields

Within:

● All content ○ My subscribed content

Limit the search to:

Items published between All and All

Article Types All Types

From http://www.emeraldinsight.com/Insight/

What:

Provide users with an advanced search interface that allows them to enter and execute advanced searches.

Use When:

When an IR system is intended for a large number of users and the skill level of the users is not known it is appropriate to provide both a general search and an advanced search. If there will be advanced or expert users provide an advanced search. When the users' intentions are not known it is appropriate to provide an advanced search.

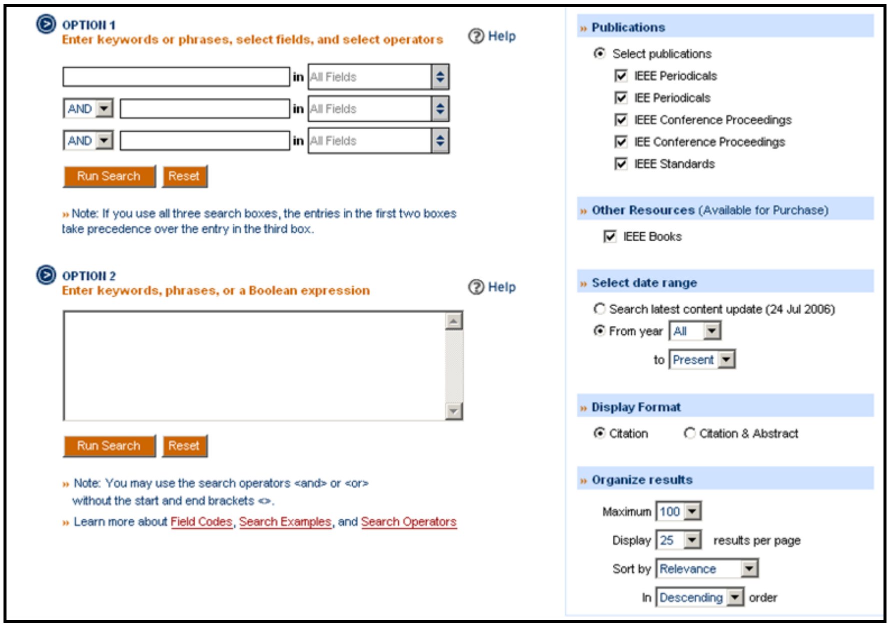
Why:

It is necessary to give more advanced users the option of using an advanced search when they would like. It is also necessary to allow users to specifically limit and enter what they are looking for by using an advanced search.

How:

Provide an advanced search interface that allows users to input terms or select options for the fields in an advanced search. Provide options for entering advanced searches. Provide an advanced search interface both with and without advanced search field codes. Provide an advanced search interface that helps users with little experience enter and execute advanced searches. Provide an advanced search that allows advanced and expert users to execute complex queries by using field codes.

Examples:



OPTION 1
Enter keywords or phrases, select fields, and select operators

[ ] in All Fields
AND [ ] in All Fields
AND [ ] in All Fields

» Note: If you use all three search boxes, the entries in the first two boxes take precedence over the entry in the third box.

OPTION 2
Enter keywords, phrases, or a Boolean expression

» Note: You may use the search operators <and> or <or> without the start and end brackets <>.
» Learn more about Field Codes, Search Examples, and Search Operators

» Publications
● Select publications
☑ IEEE Periodicals
☑ IEE Periodicals
☑ IEEE Conference Proceedings
☑ IEE Conference Proceedings
☑ IEEE Standards

» Other Resources (Available for Purchase)
☑ IEEE Books

» Select date range
○ Search latest content update (24 Jul 2006)
● From year All
to Present

» Display Format
● Citation ○ Citation & Abstract

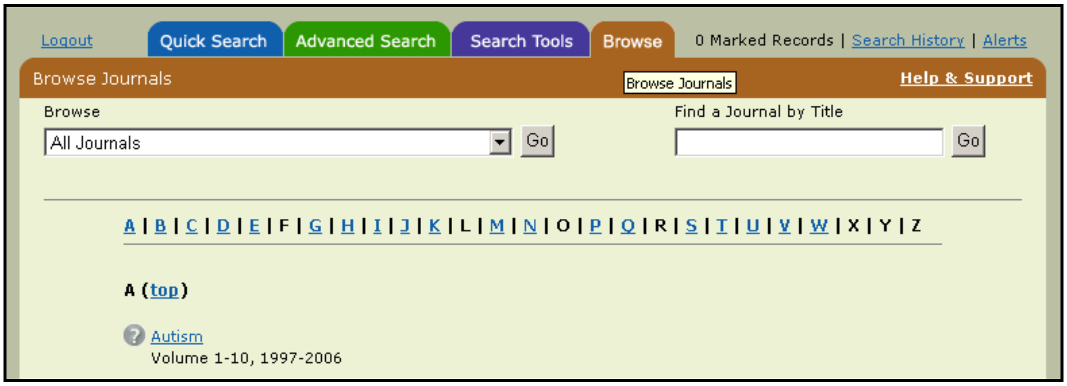
» Organize results
Maximum 100
Display 25 results per page
Sort by Relevance
In Descending order

From http://ieeexplore.ieee.org/Xplore/dynhome.jsp

How this pattern is related to others:

This pattern is part of **Information Retrieval Systems**. This pattern is related to **Advanced search link**. This pattern contains **Advanced search help**. This pattern is related to **Display Results** and **Refine Search**. This pattern contains **Search Box and button**, **Advanced Search fields**, **Execute search by clicking GO or SEARCH**, **Advanced search without codes**, and **Advanced search with codes**.

Browse


Logout
Quick Search
Advanced Search
Search Tools
Browse
0 Marked Records | Search History | Alerts
Browse Journals
Browse Journals
Help & Support
Find a Journal by Title
Go
Go
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z
A (top)

Browse
All Journals

Autism
Volume 1-10, 1997-2006

From http://www.csa.com/csaillumina/login.php

What:

Allow users to look through the contents of the system without having to specify a search query.

Use When:

When a system contains records that can be organized in some fashion (whether it is by source, publication year, keywords, etc.) allow users to look through the contents of the system by one or more of these organizing factors (for example browse by title, author, etc).

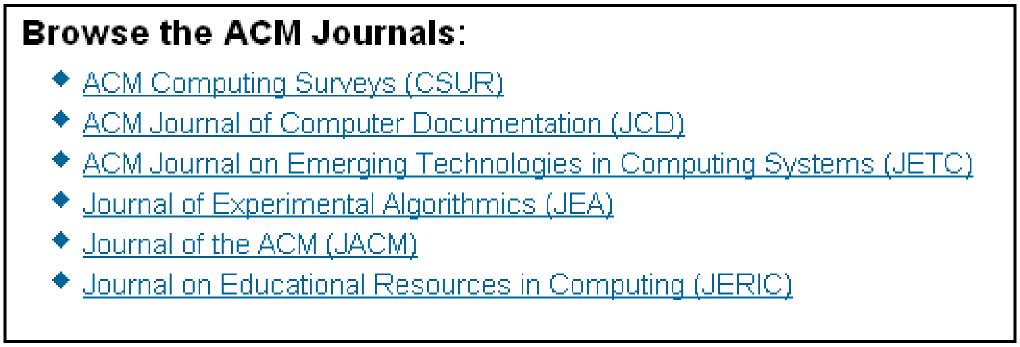
Why:

Users may have a general interest in an area and they may like to look through the content to get an idea of what types of records the system contains. For example a user may want to look through the contents of a journal to get a sense of the types of articles the journal publishes. Users may not always know exactly what they are looking for so they should be allowed to look through the contents of the system without specifying a query.

How:

Provide a Browse function in the information retrieval interface that allows users to browse the contents of the system. Provide options for browsing by source, title, publication year, etc. Also provide an option for browsing an alphabetical listing.

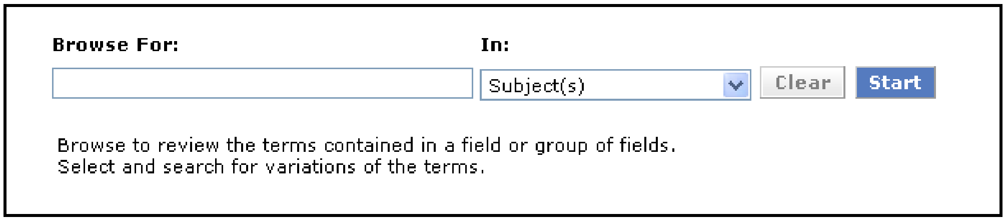
Examples:



Browse the ACM Journals:

- ACM Computing Surveys (CSUR)
- ACM Journal of Computer Documentation (JCD)
- ACM Journal on Emerging Technologies in Computing Systems (JETC)
- Journal of Experimental Algorithmics (JEA)
- Journal of the ACM (JACM)
- Journal on Educational Resources in Computing (JERIC)

From http://portal.acm.org/dl.cfm



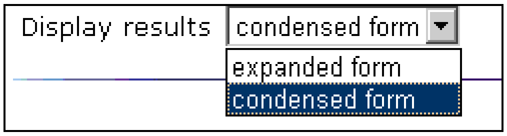
Browse to review the terms contained in a field or group of fields.
Select and search for variations of the terms.

From http://vnweb.hwwilsonweb.com/hww/browse/browse#formTop

How this pattern is related to others:

This pattern is a part of **IR systems**. This pattern contains **Results Navigation**. This pattern is related to **Display Results**.

Display Format



From http://portal.acm.org/dl.cfm

What:

Provide users with the option to toggle between a short format or long format for displaying the results.

Use When:

When displaying the results of a query to the user include allow users to choose how they would like the results displayed (with more information or less information presented).

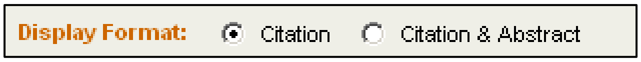
Why:

In order to provide flexibility give users the option to easily change between a long or short form in order to allow flexibility in viewing the results set. In some cases user may want to only view the citation (title, author, source, etc. of a record). In other cases a user may want to view the citation in addition to viewing the abstract. If a user know exactly what they are looking for they may want to view results in a condensed form so that they can easily and quickly browse through a lot of results. When a user does not know exactly what they are looking for they may want to be able to read the article abstract when looking through the results.

How:

Provide users with the option to select either a condensed format or an expanded format when viewing the results. The short format may include just the author, title, year, and source. The long format may include all of that information and an abstract and information about the authors.

Example:

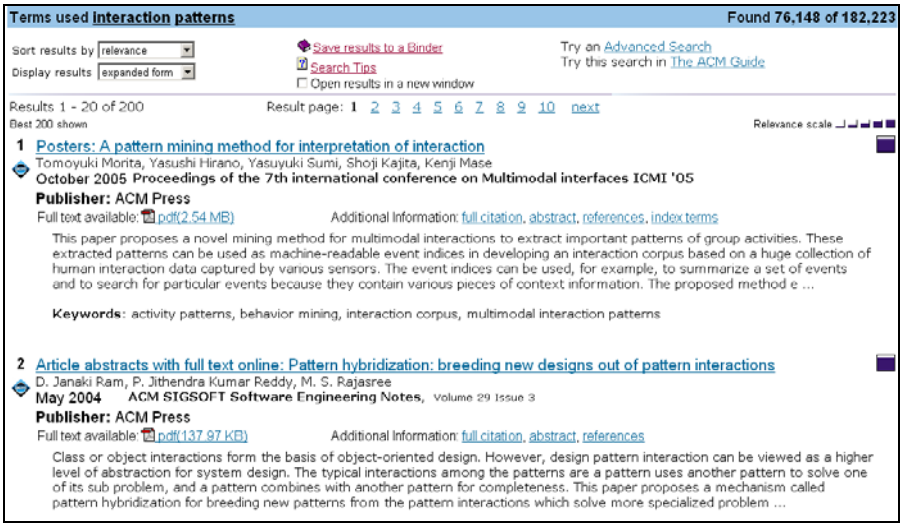


From http://ieeexplore.ieee.org/Xplore/dynhome.jsp

How this pattern is related to others:

This pattern is a part of **Display Results**. This pattern is related to **Result Set** and **Result Records**.

Display Results



Terms used Interaction patterns | Found 76,148 of 182,223

Sort results by relevance
Display results expanded form

Save results to a Binder
Search Tips
□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 | Result page: 1 2 3 4 5 6 7 8 9 10 next
Best 200 shown

1 Posters: A pattern mining method for interpretation of interaction
Tomoyuki Morita, Yasushi Hirano, Yasuyuki Sumi, Shoji Kajita, Kenji Mase
October 2005 Proceedings of the 7th international conference on Multimodal interfaces ICMI '05
Publisher: ACM Press
Full text available: pdf(2.54 MB) Additional Information: full citation, abstract, references, index terms

This paper proposes a novel mining method for multimodal interactions to extract important patterns of group activities. These extracted patterns can be used as machine-readable event indices in developing an interaction corpus based on a huge collection of human interaction data captured by various sensors. The event indices can be used, for example, to summarize a set of events and to search for particular events because they contain various pieces of context information. The proposed method e ...

Keywords: activity patterns, behavior mining, interaction corpus, multimodal interaction patterns

2 Article abstracts with full text online: Pattern hybridization: breeding new designs out of pattern interactions
D. Janaki Ram, P. Jithendra Kumar Reddy, M. S. Rajasree
May 2004 ACM SIGSOFT Software Engineering Notes, Volume 29 Issue 3
Publisher: ACM Press
Full text available: pdf(137.97 KB) Additional Information: full citation, abstract, references

Class or object interactions form the basis of object-oriented design. However, design pattern interaction can be viewed as a higher level of abstraction for system design. The typical interactions among the patterns are a pattern uses another pattern to solve one of its sub problem, and a pattern combines with another pattern for completeness. This paper proposes a mechanism called pattern hybridization for breeding new patterns from the pattern interactions which solve more specialized problem ...

From http://portal.acm.org/dl.cfm

What:

The results of a user's query need to be returned to the user in an ordered, numbered list. The user needs to be able to navigate through the results. The user should be informed of the total number of results retrieved by the system. The user should be presented with options for sorting, organizing, and displaying the results of a query.

Use When:

When a user executes a query the results of the query should be given to the user in a default display format.

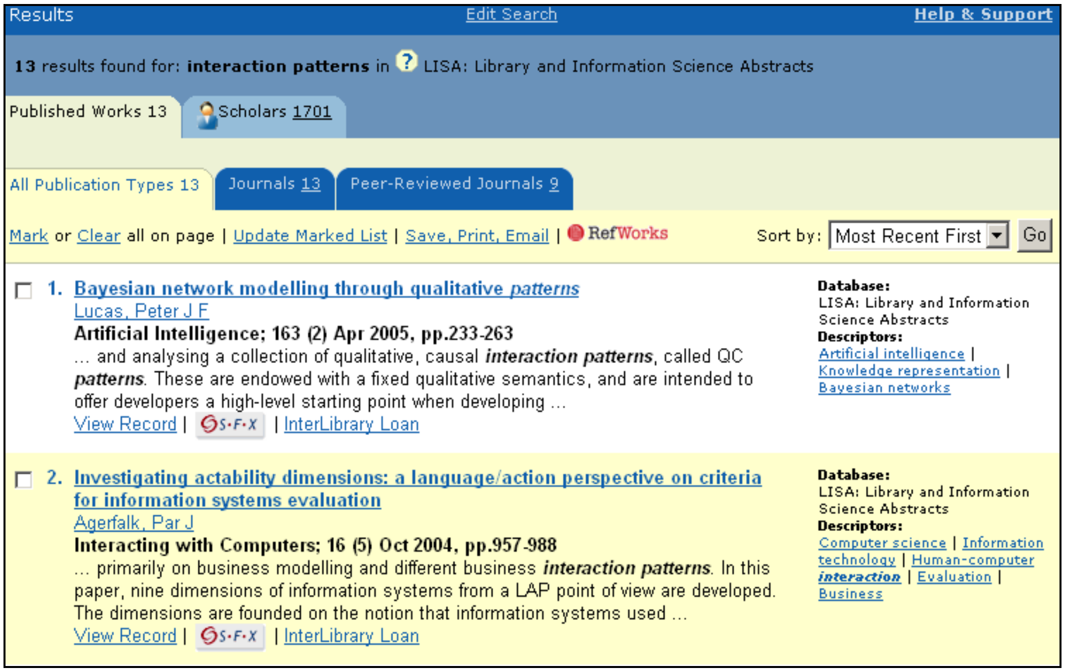
Why:

The user's goal is to retrieve information related to their search query therefore the system must present the results of the query to the user and allow the user to navigate through the results.

How:

The results of a user's query should be presented to the user in the default display format unless the user has specified another display format. The results should be presented in an ordered, numbered list. The user should be informed of the total number of results a query retrieved. The user should be able to navigate through the results of a query. The user should be presented with options for sorting, organizing, and displaying the results of a query.

Examples:



13 results found for: **interaction patterns** in ? LISA: Library and Information Science Abstracts

□ 1. Bayesian network modelling through qualitative *patterns*
Lucas, Peter J F
Artificial Intelligence; 163 (2) Apr 2005, pp.233-263
... and analysing a collection of qualitative, causal ***interaction patterns***, called QC ***patterns***. These are endowed with a fixed qualitative semantics, and are intended to offer developers a high-level starting point when developing ...
View Record | S·F·X | InterLibrary Loan

Database: LISA: Library and Information Science Abstracts
Descriptors: Artificial intelligence | Knowledge representation | Bayesian networks

□ 2. Investigating actability dimensions: a language/action perspective on criteria for information systems evaluation
Agerfalk, Par J
Interacting with Computers; 16 (5) Oct 2004, pp.957-988
... primarily on business modelling and different business ***interaction patterns***. In this paper, nine dimensions of information systems from a LAP point of view are developed. The dimensions are founded on the notion that information systems used ...
View Record | S·F·X | InterLibrary Loan

Database: LISA: Library and Information Science Abstracts
Descriptors: Computer science | Information technology | Human-computer ***interaction*** | Evaluation | Business

http://www.csa.com/csaillumina/login.php

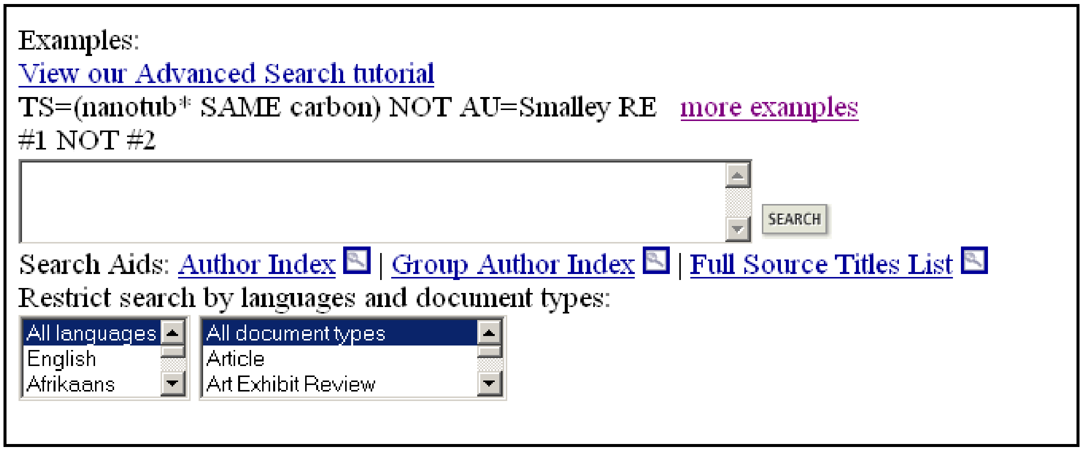
How this pattern is related to others:

This pattern is part of **Information Retrieval Systems**.

This pattern contains **No results**, **Result Set**, **Search History**, **Sort By**, **Sorting Order**, **Display Format**, **Mark Documents**, **Number of Results per Page**, **Total Number of Retrieved Results**, and **Results Navigation**.

This pattern is related to **Browse**, **Advanced Search** and **Refine Search**. This pattern is also related to **Execute Search by clicking SEARCH** and **Execute Search on ENTER**.

Execute search by clicking GO or Search (Do not execute on ENTER)



Examples:
View our Advanced Search tutorial
TS=(nanotub* SAME carbon) NOT AU=Smalley RE more examples
#1 NOT #2

Search Aids: Author Index | Group Author Index | Full Source Titles List
Restrict search by languages and document types:

From http://apps.isiknowledge.com/

What:

Execute a search query entered by the user only after the user clicks go or search (do not execute on ENTER)

Use When:

When the time to recover from a mistake may be unacceptable to the user and/or when the likelihood of the user making a mistake is increased (as in an advanced search, especially advanced search options that allow user to enter multiple commands in the same field).

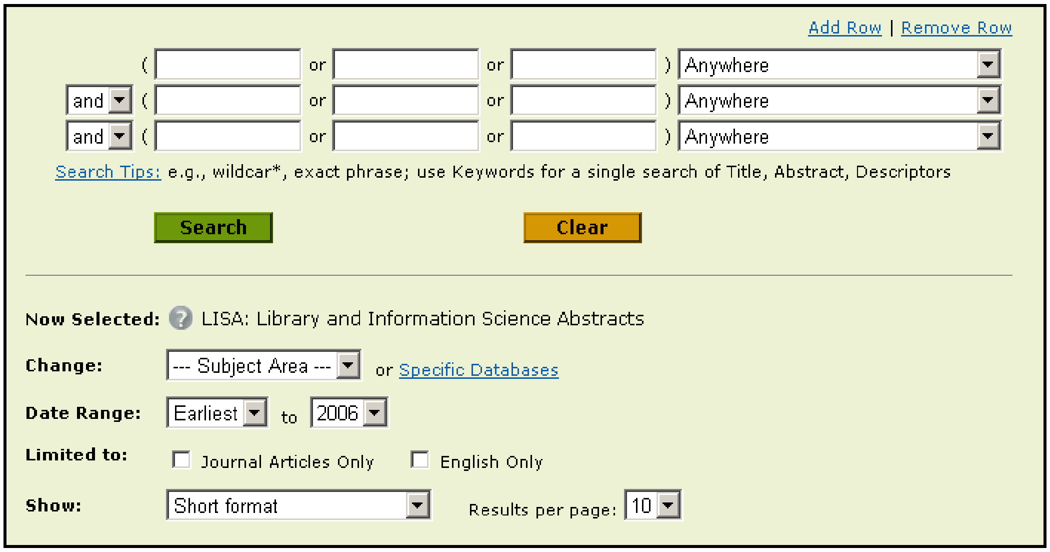
Why:

When an information retrieval system may take a considerable amount of time to execute a query and return the results to a user the search should be executed by clicking GO or Search (not by pressing enter) because the likelihood of the user clicking GO or Search with the mouse in error is much less than the likelihood of the user pressing ENTER in error. If the time it takes to return the results and recover from an error may be unacceptable to the user then the search should be executed by clicking GO or Search rather than simply pressing enter.

How:

Provide functionality that forces the user to click a GO or Search button with the mouse before executing a query rather than simply allowing the user to press ENTER to execute a query (this means that pressing ENTER does nothing).

Examples:



(or or) Anywhere
and (or or) Anywhere
and (or or) Anywhere

Search Tips: e.g., wildcar*, exact phrase; use Keywords for a single search of Title, Abstract, Descriptors

Now Selected: LISA: Library and Information Science Abstracts

Change: --- Subject Area --- or Specific Databases

Date Range: Earliest to 2006

Limited to: □ Journal Articles Only □ English Only

Show: Short format Results per page: 10

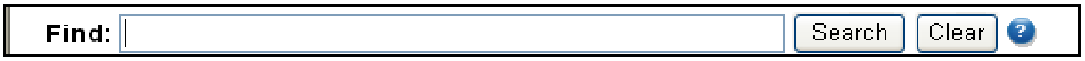
From http://www.csa.com/csaillumina/login.php

How this pattern is related to others:

This pattern is a part of **General search** and **Advanced search**. This pattern is also a part of **Search Box and Button**.

This pattern is related to **Display Results**.

Execute Search on ENTER (or by clicking GO or Search)



From http://www.ebscohost.com/

What:

Execute a search query entered by the user after the user presses the ENTER key (or clicks go or search).

Use When:

When an IR system can execute a user's query and return the results relatively quickly. This may be more appropriate for a general search query interface but may also be acceptable in an advanced search query interface.

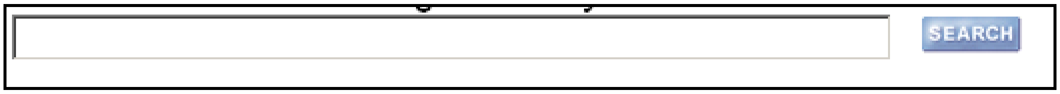
Why:

Executing a search query after a user presses ENTER saves the user the extra time it takes to move the mouse to click a button in the interface. If a user can recover from an error made when entering a search query in a reasonable amount of time this is appropriate. If the system takes a reasonably short time to execute a query and return the results this is appropriate because the time that may be wasted in correcting a mistake will be minimal or acceptable to the user. If the user is entering a general search this may be more appropriate because the likelihood of making a mistake in entering a general search query is much less than the likelihood of making a mistake in entering an advance search query. This is because of the number of options available in a general search (most likely one field) is fewer than the possible number of options in an advanced search (possibly many fields to fill in).

How:

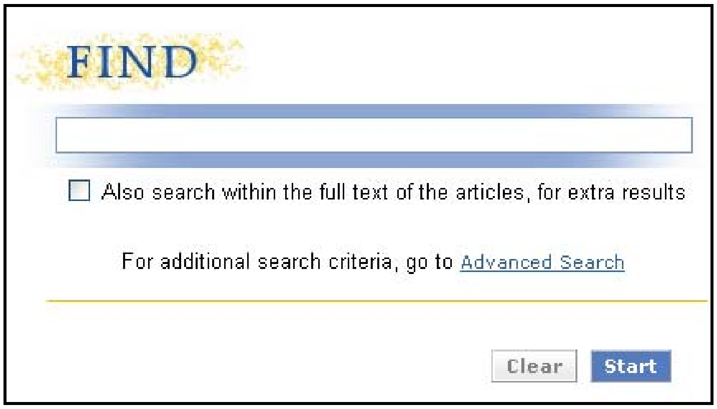
Provide functionality that allows the user to simply press the ENTER key (or click GO or Search) to execute a query rather than only executing a query when the user clicks GO or Search with the mouse.

Example:



From http://portal.acm.org/dl.cfm

Example:



□ Also search within the full text of the articles, for extra results

For additional search criteria, go to Advanced Search

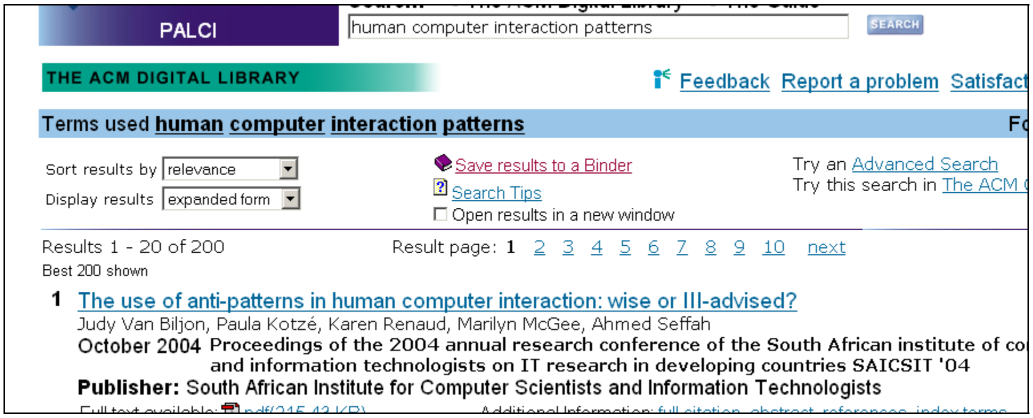
From http://vnweb.hwwilsonweb.com/hww/login.jhtml

How this pattern is related to others:

This pattern is a part of **General search** and **Search Box and Button**.

This pattern is also related to **Display results**.

Executed Search query displayed in search box



human computer interaction patterns

Terms used **human computer interaction patterns**

Sort results by relevance

Display results expanded form

□ Open results in a new window

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

1 The use of anti-patterns in human computer interaction: wise or Ill-advised?
Judy Van Biljon, Paula Kotzé, Karen Renaud, Marilyn McGee, Ahmed Seffah
October 2004 Proceedings of the 2004 annual research conference of the South African institute of co and information technologists on IT research in developing countries SAICSIT '04
Publisher: South African Institute for Computer Scientists and Information Technologists

From http://portal.acm.org/dl.cfm

What:

A search query entered by a user should always be displayed (the way the user entered it) in the search box in the top (or both top and bottom) portion of the interface in which the results of the search query are displayed.

Use When:

When the results of a search query are displayed the search query used to obtain those results should be displayed.

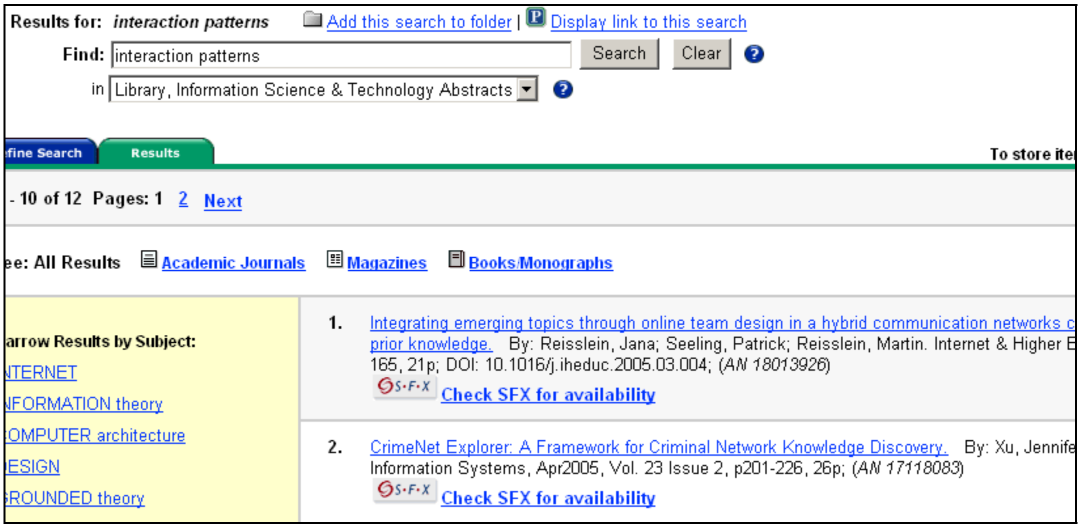
Why:

Providing the users' already executed search query in the search box reminds the user of the exact search query they have executed. Providing a search box with the user's current search also allows the user to easily modify the search with little effort at any time while browsing through the results. The user can simply modify the current search on the results page (for example if they made a typo) without having to return to a search page to modify the search query.

How:

When displaying the results of a query the users query should always be displayed in a search box located on the top (or both top and bottom) of the interface. The search query should appear exactly as the user input it (this means NOT including tags that the system uses to execute a query).

Examples:



Results for: *interaction patterns* Add this search to folder | Display link to this search

Find: interaction patterns Search Clear

in Library, Information Science & Technology Abstracts

- 10 of 12 Pages: 1 2 Next

ee: All Results Academic Journals Magazines Books/Monographs

arrow Results by Subject:
NTERNET
NFORMATION theory
OMPUTER architecture
ESIGN
ROUNDED theory

1. Integrating emerging topics through online team design in a hybrid communication networks c prior knowledge. By: Reisslein, Jana; Seeling, Patrick; Reisslein, Martin. Internet & Higher E 165, 21p; DOI: 10.1016/j.iheduc.2005.03.004; (*AN 18013926*)
Check SFX for availability

2. CrimeNet Explorer: A Framework for Criminal Network Knowledge Discovery. By: Xu, Jennife Information Systems, Apr2005, Vol. 23 Issue 2, p201-226, 26p; (*AN 17118083*)
Check SFX for availability

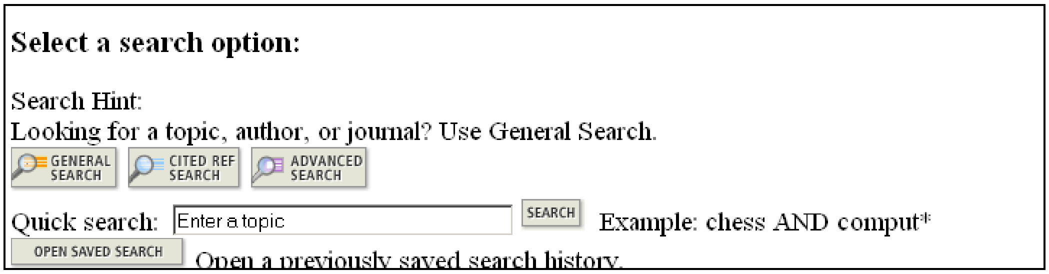
From http://www.ebscohost.com/

How this pattern is related to others:

This pattern is a part of **Display results**.

This pattern is related to **Search box and button**.

General Search Example



Select a search option:

Search Hint:
Looking for a topic, author, or journal? Use General Search.

Quick search: Enter a topic

Example: chess AND comput*

Open a previously saved search history.

From http://apps.isiknowledge.com/

What:

Provide an example of a general search near the search box to illustrate the types of searches that can be done using a general search.

Use When:

When the information retrieval system contains a general search it is appropriate to provide examples that illustrate the types of general searches that can be performed.

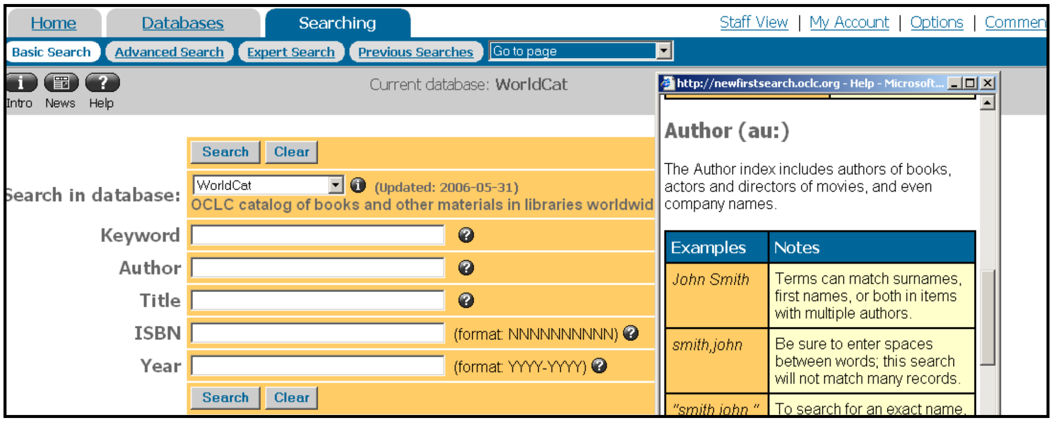
Why:

Not all users will be familiar with how to generate queries therefore it is appropriate to show users the types of queries that can be performed using the system. In addition not all systems conform to the same rules for using things like AND, OR, NOT, etc. therefore by providing examples this information is quickly communicated with the users.

How:

Provide an example of a general search directly in the general search interface. In addition provide more examples that may be accessed through the general search help. Provide a link on the general search interface that takes the users to more examples that illustrate how to use the most common features of a general search. The link that takes users to more examples should open a new window that displays the examples. This should be done in order to allow the user to view the examples while entering their query.

Examples:



Current database: WorldCat

WorldCat (Updated: 2006-05-31)
OCLC catalog of books and other materials in libraries worldwid

ISBN (format: NNNNNNNNNN)
Year (format: YYYY-YYYY)

Author (au:)

The Author index includes authors of books, actors and directors of movies, and even company names.

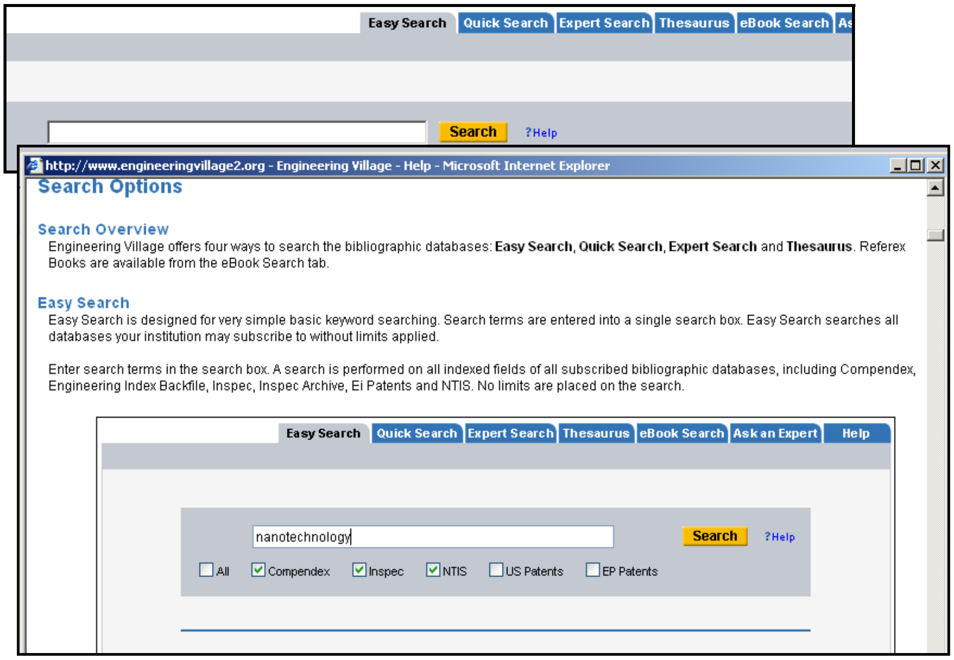
Examples	Notes
John Smith	Terms can match surnames, first names, or both in items with multiple authors.
smith,john	Be sure to enter spaces between words; this search will not match many records.
"smith john "	To search for an exact name,

From http://newfirstsearch.oclc.org/

How this pattern is related to others:

This pattern is a part of **General search** and **General search help**.

General Search Help



Search Options

Search Overview

Engineering Village offers four ways to search the bibliographic databases: **Easy Search**, **Quick Search**, **Expert Search** and **Thesaurus**. Referex Books are available from the eBook Search tab.

Easy Search

Easy Search is designed for very simple basic keyword searching. Search terms are entered into a single search box. Easy Search searches all databases your institution may subscribe to without limits applied.

Enter search terms in the search box. A search is performed on all indexed fields of all subscribed bibliographic databases, including Compendex, Engineering Index Backfile, Inspec, Inspec Archive, Ei Patents and NTIS. No limits are placed on the search.

nanotechnology

□ All ☑ Compendex ☑ Inspec ☑ NTIS □ US Patents □ EP Patents

From http://www.engineeringvillage2.org/

What:

Provide Help that is appropriate for general searches. The General Search Help should contain information that aids users in constructing, entering, and executing general search queries. In addition it should provide general information about using the information retrieval system. The general search help should contain examples, tips, and a tutorial.

Use When:

When an Information Retrieval System has a general search it is necessary to provide general search help. When there may be users with varying levels of search experience it is appropriate to provide information that will allow them to take advantage of the general search features within the system.

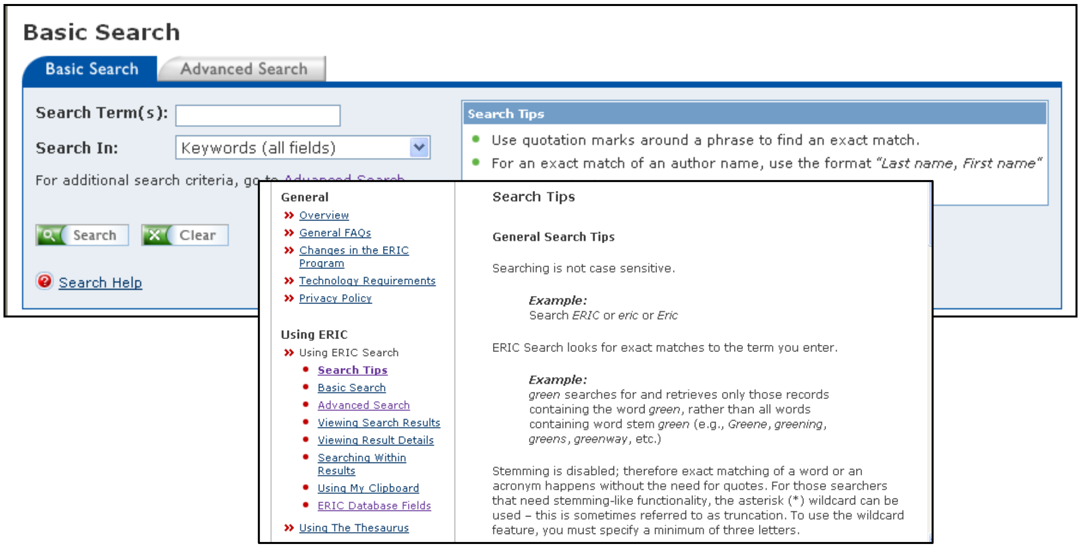
Why:

We would like to provide help to those trying to execute general searches. New or novice users may need help executing a general search therefore it is appropriate to provide a help function. In addition not all systems conform to the same rules for using things like AND, OR, NOT, etc. therefore this system specific information should be communicated to the user in the help.

How:

Provide a link to the general search help near the search box. Provide information that would aid users in constructing a general search. Be sure that the help appears in a separate window so that the users can view the help information and enter their queries at the same time. The general search help should contain examples of general searches, tips for constructing queries, and a tutorial with in depth information about the general search.

Examples:



For additional search criteria, go to Advanced Search

Search Tips

- Use quotation marks around a phrase to find an exact match.
- For an exact match of an author name, use the format *"Last name, First name"*

Search Tips

General Search Tips

Searching is not case sensitive.

Example:
Search *ERIC* or *eric* or *Eric*

ERIC Search looks for exact matches to the term you enter.

Example:
green searches for and retrieves only those records containing the word *green*, rather than all words containing word stem *green* (e.g., *Greene*, *greening*, *greens*, *greenway*, etc.)

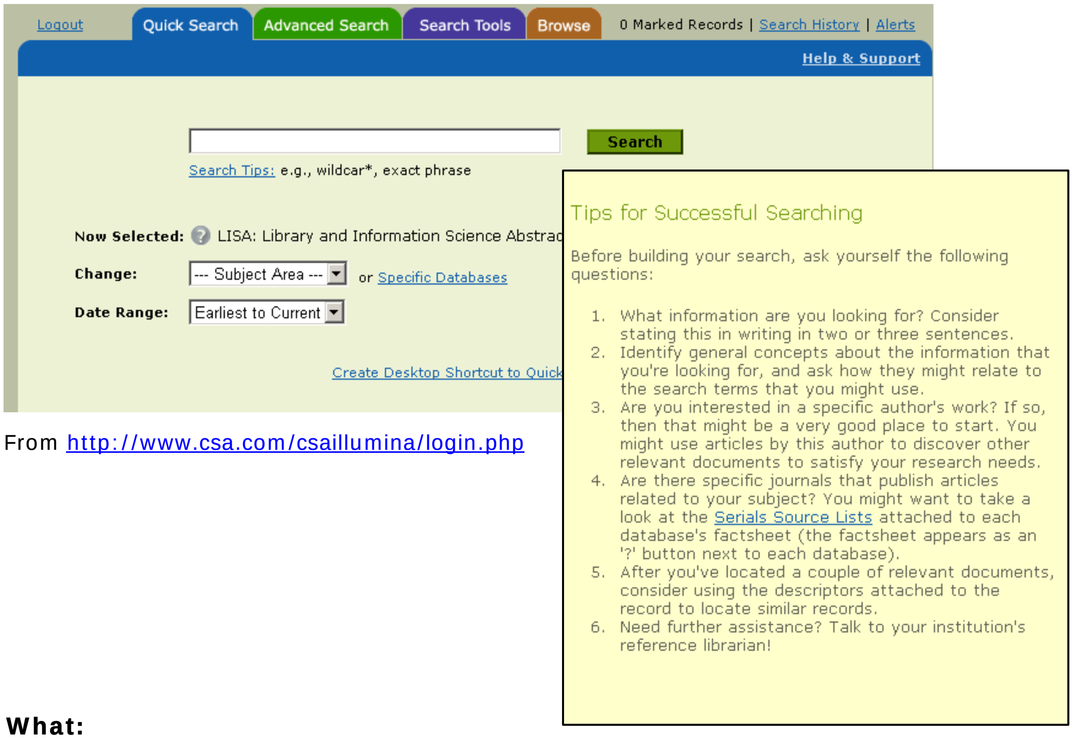
Stemming is disabled; therefore exact matching of a word or an acronym happens without the need for quotes. For those searchers that need stemming-like functionality, the asterisk (*) wildcard can be used – this is sometimes referred to as truncation. To use the wildcard feature, you must specify a minimum of three letters.

From http://www.eric.ed.gov/ERICWebPortal/Home.portal

How this pattern is related to others:

This pattern is a part of **Information Retrieval Systems**, **Help** and **General search**. This pattern contains **General search examples**, and **General search tips**. This pattern is related to **Help window**.

General Search Tips


Logout
Quick Search
Advanced Search
Search Tools
Browse
0 Marked Records | Search History | Alerts
Help & Support
Search
Create Desktop Shortcut to Quick

Search Tips: e.g., wildcar*, exact phrase

Now Selected: LISA: Library and Information Science Abstrac

Change: --- Subject Area --- or Specific Databases

Date Range: Earliest to Current

Tips for Successful Searching

Before building your search, ask yourself the following questions:

1. What information are you looking for? Consider stating this in writing in two or three sentences.
2. Identify general concepts about the information that you're looking for, and ask how they might relate to the search terms that you might use.
3. Are you interested in a specific author's work? If so, then that might be a very good place to start. You might use articles by this author to discover other relevant documents to satisfy your research needs.
4. Are there specific journals that publish articles related to your subject? You might want to take a look at the Serials Source Lists attached to each database's factsheet (the factsheet appears as an '?' button next to each database).
5. After you've located a couple of relevant documents, consider using the descriptors attached to the record to locate similar records.
6. Need further assistance? Talk to your institution's reference librarian!

From http://www.csa.com/csaillumina/login.php

What:

Provide quick tips for general searching in the general search interface. Provide Tips in the General search help also.

Use When:

When trying to provide quick help in the general search interface it is appropriate to provide tips. When a system contains a general search and general search help this is appropriate.

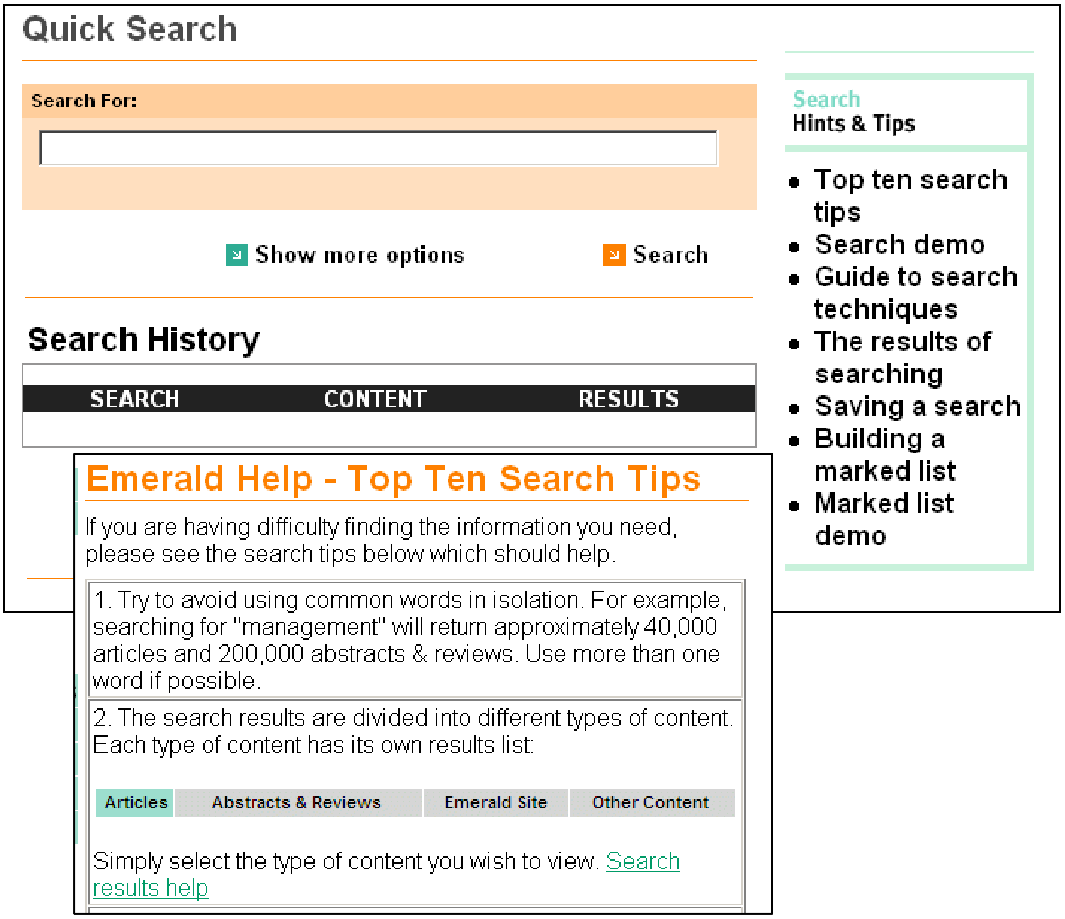
Why:

In order to help users, that may not be familiar with the system, get started using the system provide search tips in the general search interface. For those users that do not have time to read the help documentation this provides quick help directly in the interface that can help a user get started without spending a lot of time reading.

How:

Provide quick tips in the general search interface that can help a user get started or provide a link to general search tips in the general search interface. If a link is provided be sure that the tips open in a new window so that the user can view the tips and enter a query at the same time. Also provide access to the tips from the general search help.

Examples:



Emerald Help - Top Ten Search Tips

If you are having difficulty finding the information you need, please see the search tips below which should help.

1. Try to avoid using common words in isolation. For example, searching for "management" will return approximately 40,000 articles and 200,000 abstracts & reviews. Use more than one word if possible.

2. The search results are divided into different types of content. Each type of content has its own results list:

Simply select the type of content you wish to view. Search results help

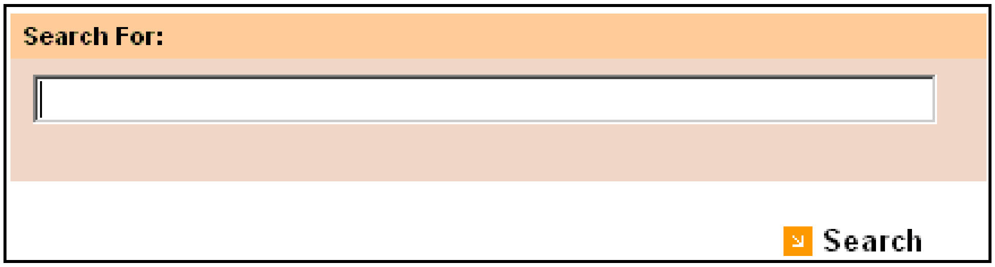
From http://www.emeraldinsight.com/Insight/

How this pattern is related to others:

This pattern is a part of **General Search** and **General Search Help**.

This pattern is related to **General Search Example**.

General Search



From http://www.emeraldinsight.com/Insight/

What:

Provide users with a general search.

Use When:

When an IR systems is intended for a large number of users and the skill level of the users is not known or is varied and the intentions of the users are not known provide a general search.

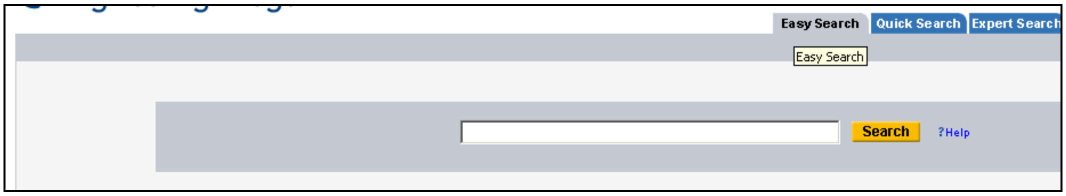
Why:

Many searches will be broad, general searches therefore it is appropriate to provide a general search on the initial interface. Many users may prefer to perform simple searches. Also for novice users or quick, broad searches it is appropriate to provide a general search.

How:

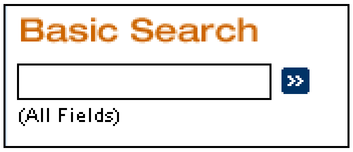
Provide a general search with one text box and a search button that enables users to simply and quickly input a text query on the initial interface.

Examples:



From http://www.engineeringvillage2.org/

Examples:



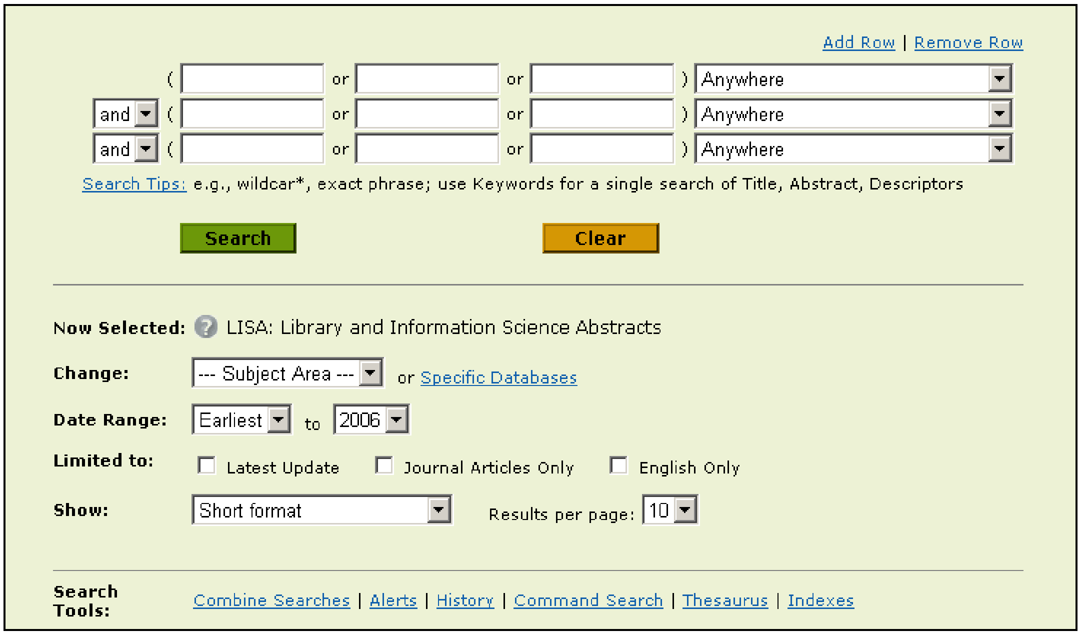
From http://ieeexplore.ieee.org/Xplore/dynhome.jsp

How this pattern is related to others:

This pattern is part of **IR Systems**. This pattern is related to **Refine Search**.

This pattern contains **Advanced search link**, **Search Box and button**, **Execute search on ENTER**, **Execute Search by clicking SEARCH**, **Help**, **General Search Help**, **General Search Examples**, and **General Search Tips**.

Grouping Advanced Search Fields


Add Row | Remove Row
(or or) Anywhere
and (or or) Anywhere
and (or or) Anywhere
Search
Clear
Change: --- Subject Area --- or Specific Databases
Limited to: □ Latest Update □ Journal Articles Only □ English Only
Search Tools: Combine Searches | Alerts | History | Command Search | Thesaurus | Indexes

Search Tips: e.g., wildcar*, exact phrase; use Keywords for a single search of Title, Abstract, Descriptors

Now Selected: LISA: Library and Information Science Abstracts

Date Range: Earliest to 2006

Show: Short format Results per page: 10

From http://www.csa.com/csaillumina/login.php

What:

Visually group advanced search fields to indicate similarity and relationships between fields.

Use When:

When an advanced search contains many advanced search fields this is appropriate.

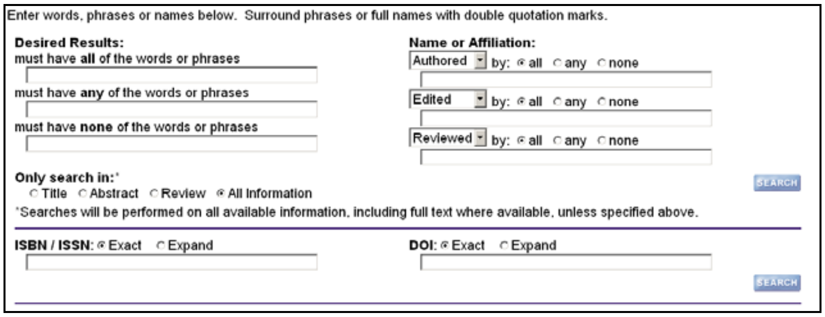
Why:

Visually separating or grouping the advanced search fields helps communicate to the user what fields are related and differentiate them from the fields that are not related.

How:

Group advanced search fields that are similar with one another and place dissimilar fields in different groups. Group and separate fields by using lines, sections, colors, etc.

Examples:



Enter words, phrases or names below. Surround phrases or full names with double quotation marks.

Desired Results:
must have **all** of the words or phrases
must have **any** of the words or phrases
must have **none** of the words or phrases

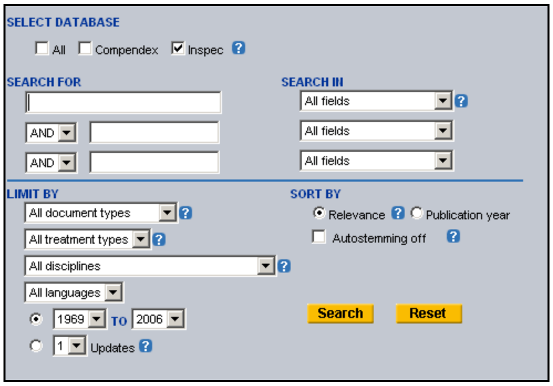
Name or Affiliation:
Authored by: ● all ○ any ○ none
Edited by: ● all ○ any ○ none
Reviewed by: ● all ○ any ○ none

Only search in:*
○ Title ○ Abstract ○ Review ● All Information
*Searches will be performed on all available information, including full text where available, unless specified above.

ISBN / ISSN: ● Exact ○ Expand

DOI: ● Exact ○ Expand

From http://portal.acm.org/dl.cfm



SELECT DATABASE
□ All □ Compendex ☑ Inspec

SEARCH FOR
AND
AND

SEARCH IN
All fields
All fields
All fields

LIMIT BY
All document types
All treatment types
All disciplines
All languages
● 1969 TO 2006
○ 1 Updates

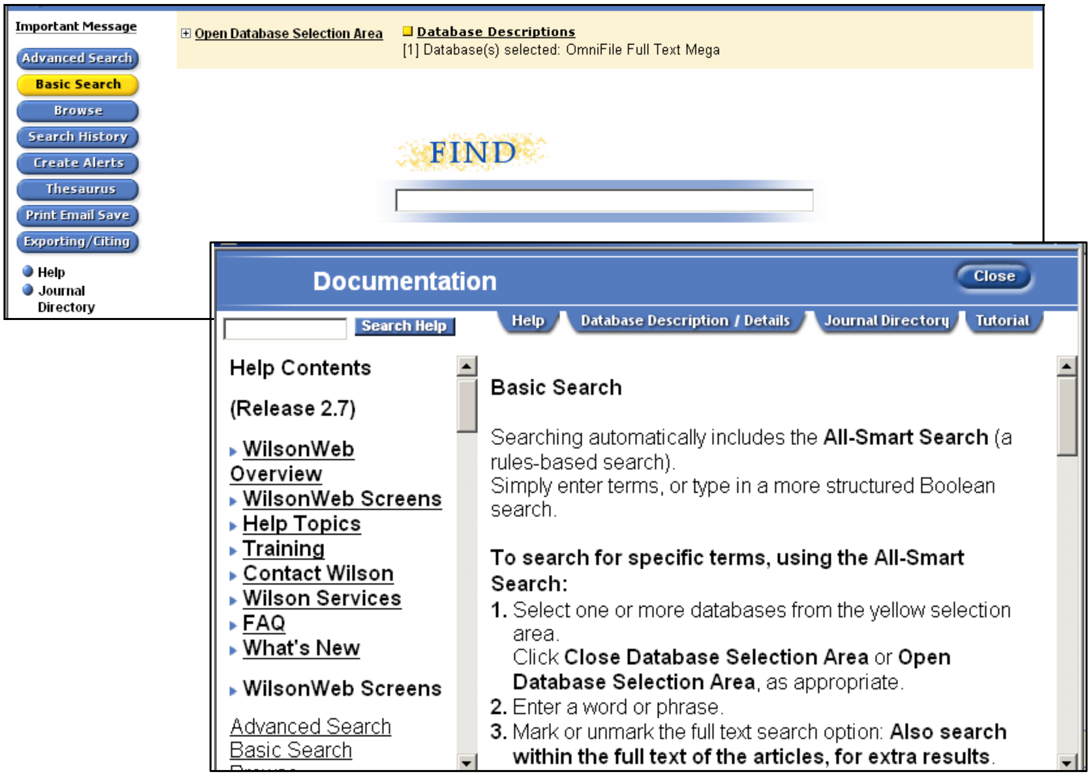
SORT BY
● Relevance ○ Publication year
□ Autostemming off

From http://www.engineeringvillage2.org/

How this pattern is related to others:

This pattern is a part of **Advanced search fields**. This pattern is related to **Tab through Search fields**.

Help Window



Basic Search

Searching automatically includes the **All-Smart Search** (a rules-based search).
Simply enter terms, or type in a more structured Boolean search.

To search for specific terms, using the All-Smart Search:

1. Select one or more databases from the yellow selection area. Click **Close Database Selection Area** or **Open Database Selection Area**, as appropriate.
2. Enter a word or phrase.
3. Mark or unmark the full text search option: **Also search within the full text of the articles, for extra results**.

From http://vnweb.hwwilsonweb.com/hww/login.jhtml

What:

Provide Help in a new window.

Use When:

When a user selects any Help option it is appropriate to provide the requested help in a new window.

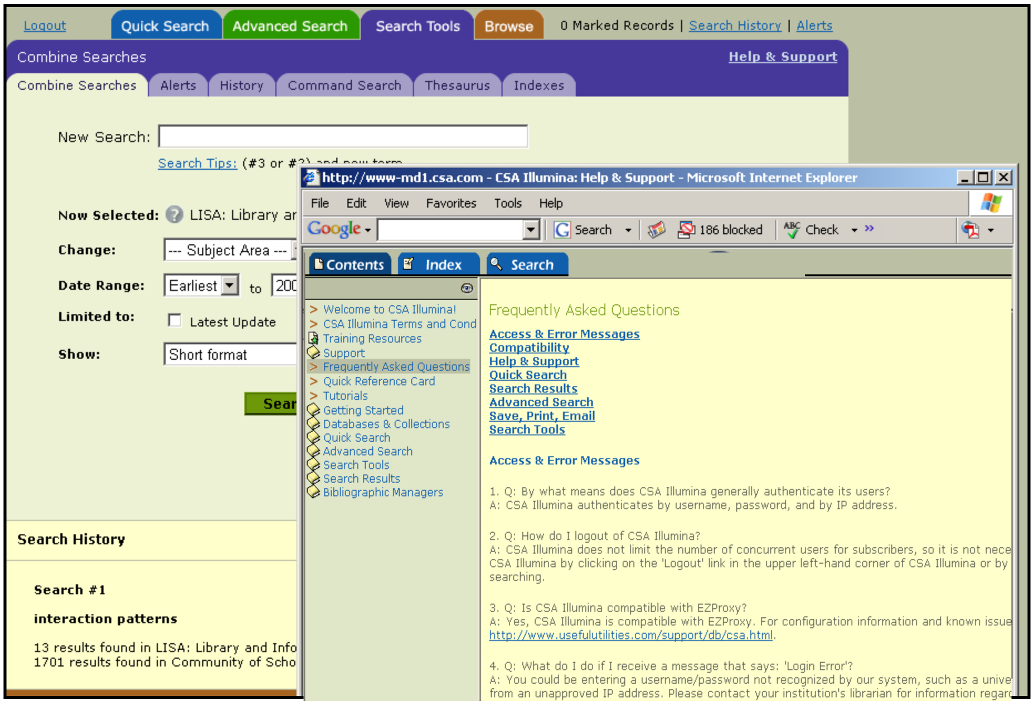
Why:

Help should be provided in a new window so that the user can view both the Help documentation and the search interface at the same time. This allows the user to read the help and follow the steps in the interface.

How:

Provide a new help window in the interface when a user selects any help option. Be sure that the Help Window does not fill the entire interface. The Help window should only take up part of the interface so that the search interface can still be seen.

Examples:



Search #1

interaction patterns

13 results found in LISA: Library and Info
1701 results found in Community of Scho

1. Q: By what means does CSA Illumina generally authenticate its users?
A: CSA Illumina authenticates by username, password, and by IP address.

2. Q: How do I logout of CSA Illumina?
A: CSA Illumina does not limit the number of concurrent users for subscribers, so it is not nece
CSA Illumina by clicking on the 'Logout' link in the upper left-hand corner of CSA Illumina or by
searching.

3. Q: Is CSA Illumina compatible with EZProxy?
A: Yes, CSA Illumina is compatible with EZProxy. For configuration information and known issue
http://www.usefulutilities.com/support/db/csa.html.

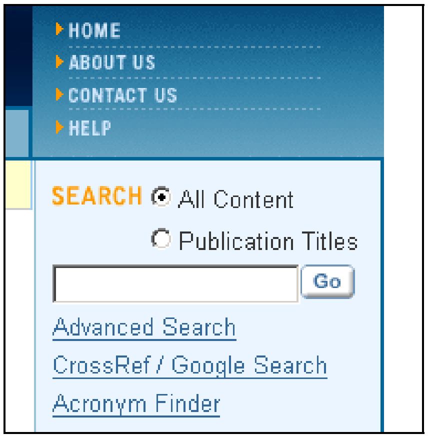
4. Q: What do I do if I receive a message that says: 'Login Error'?
A: You could be entering a username/password not recognized by our system, such as a unive
from an unapproved IP address. Please contact your institution's librarian for information regar

From http://www.csa.com/csaillumina/login.php

How this pattern is related to others:

This pattern is a part of **Help**. This pattern is related to **General Search Help** and **Advanced Search Help**.

Help



From http://www3.interscience.wiley.com/

What:

Provide a help function very close to each search box. Provide access to help from any part of the system. The Help should contain information that aids users in constructing search queries and using the system.

Use When:

Provide Help on every interface within the system. It is appropriate for users to need help at any time during the search process. Be sure that the help dialog that appears is appropriate to what the user is currently viewing for example if the user is currently looking at the advanced search interface be sure that the help dialog that appears is for advanced searches, not general searches.

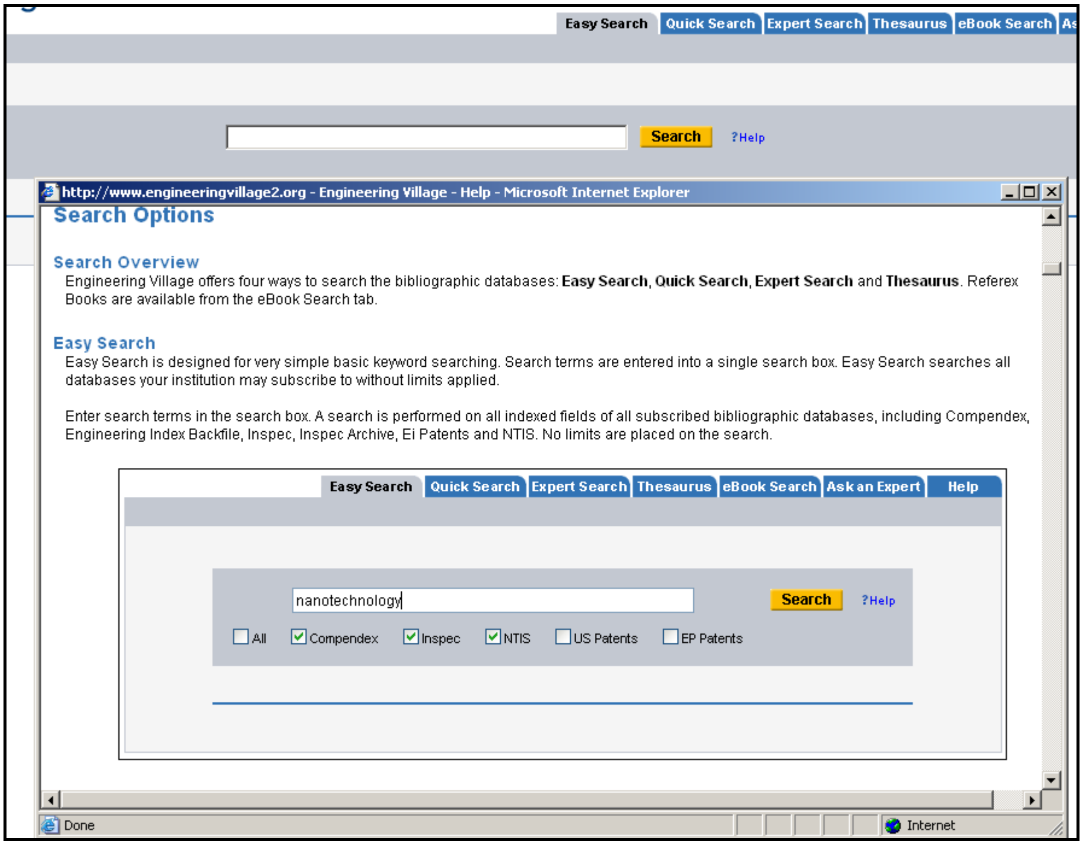
Why:

Users may need help at any point in the search process. It is therefore appropriate to provide a way to access help at any time. It should be easy to find the help therefore place the Help link close to the search box.

How:

Provide a link to the Help near the search box. Be sure that the help provided is appropriate to what the user is doing at the time, for example if the user is constructing an advanced search query the help provided should be appropriate to an advanced search but all other help documentation should also be accessible from the help window that appears. Be sure that the help appears in a separate window.

Examples:



Search Options

Search Overview

Engineering Village offers four ways to search the bibliographic databases: **Easy Search**, **Quick Search**, **Expert Search** and **Thesaurus**. Referex Books are available from the eBook Search tab.

Easy Search

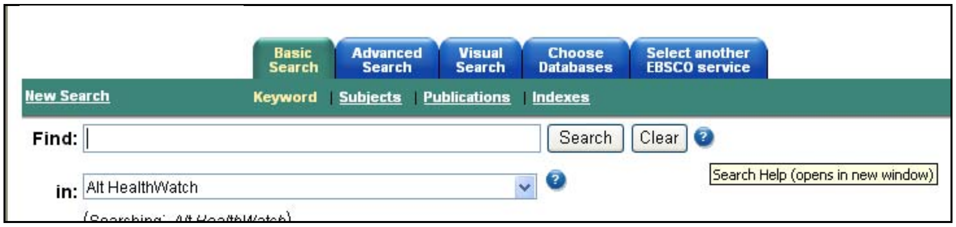
Easy Search is designed for very simple basic keyword searching. Search terms are entered into a single search box. Easy Search searches all databases your institution may subscribe to without limits applied.

Enter search terms in the search box. A search is performed on all indexed fields of all subscribed bibliographic databases, including Compendex, Engineering Index Backfile, Inspec, Inspec Archive, Ei Patents and NTIS. No limits are placed on the search.

nanotechnology

□ All ☑ Compendex ☑ Inspec ☑ NTIS □ US Patents □ EP Patents

From http://www.engineeringvillage2.org/



Find:

in: Alt HealthWatch

(Searching: Alt HealthWatch)

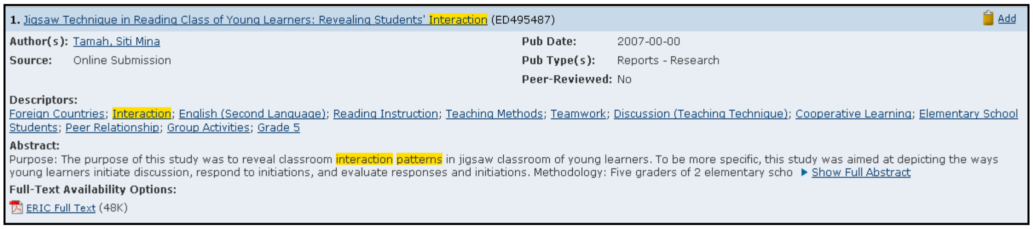
From http://www.ebscohost.com/

How this pattern is related to others:

This pattern is a part of **IR systems** and **General Search**.

This pattern contains **General search help** and **Advanced search help**. This pattern also contains **Help Window**. This pattern is related to **No Results**.

Indicate Search Terms in Result Set



1. Jigsaw Technique in Reading Class of Young Learners: Revealing Students' Interaction (ED495487) Add

Author(s): Tamah, Siti Mina
Source: Online Submission
Pub Date: 2007-00-00
Pub Type(s): Reports - Research
Peer-Reviewed: No

Descriptors:
Foreign Countries; Interaction; English (Second Language); Reading Instruction; Teaching Methods; Teamwork; Discussion (Teaching Technique); Cooperative Learning; Elementary School Students; Peer Relationship; Group Activities; Grade 5

Abstract:
Purpose: The purpose of this study was to reveal classroom interaction patterns in jigsaw classroom of young learners. To be more specific, this study was aimed at depicting the ways young learners initiate discussion, respond to initiations, and evaluate responses and initiations. Methodology: Five graders of 2 elementary scho ▸ Show Full Abstract

Full-Text Availability Options:
ERIC Full Text (48K)

From http://www.eric.ed.gov/ERICWebPortal/Home.portal

What:

The user's search terms should be indicated when they appear in the result set.

Use When:

Whenever a users' search term appears in the results it should be differentiated from the rest of the words in the result set.

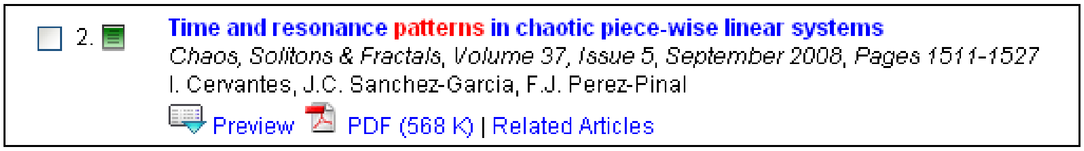
Why:

Users are presented with a lot of information after they execute a query. The user is trying to find information related to their search query therefore when the term(s) a user included in their search query appear in the result set the term(s) should be indicated or differentiated in some way from all other terms in order to draw the user's attention to the terms which were included in their query.

How:

Search terms should be indicated in the result set by differentiating them from other words in the results set. This can be done by highlighting the search terms, **bolding the search terms**, *italicizing the search terms*, **changing the color of the search terms**, etc.

Examples:

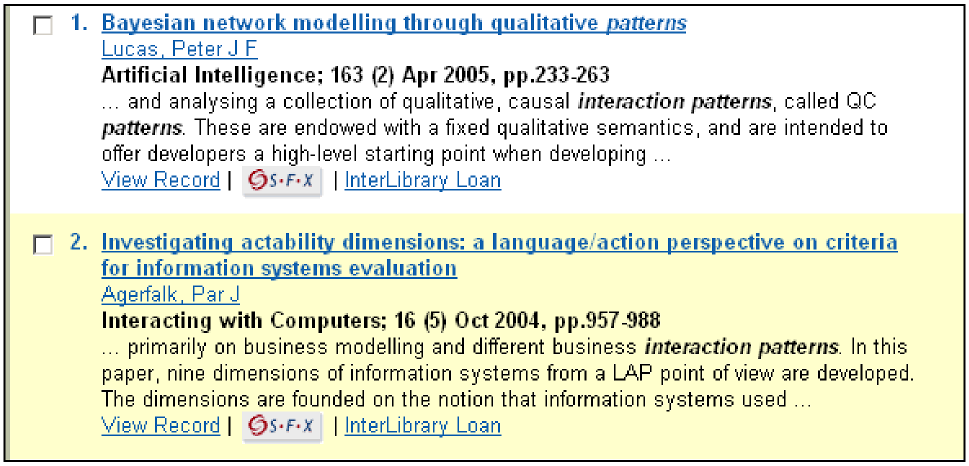


□ 2.

Time and resonance patterns in chaotic piece-wise linear systems
Chaos, Solitons & Fractals, Volume 37, Issue 5, September 2008, Pages 1511-1527
I. Cervantes, J.C. Sanchez-Garcia, F.J. Perez-Pinal
Preview PDF (568 K) | Related Articles

From http://www.blackwell-synergy.com/

Examples:



□ 1. Bayesian network modelling through qualitative *patterns*
Lucas, Peter J F
Artificial Intelligence; 163 (2) Apr 2005, pp.233-263
... and analysing a collection of qualitative, causal ***interaction patterns***, called QC ***patterns***. These are endowed with a fixed qualitative semantics, and are intended to offer developers a high-level starting point when developing ...
View Record | S•F•X | InterLibrary Loan

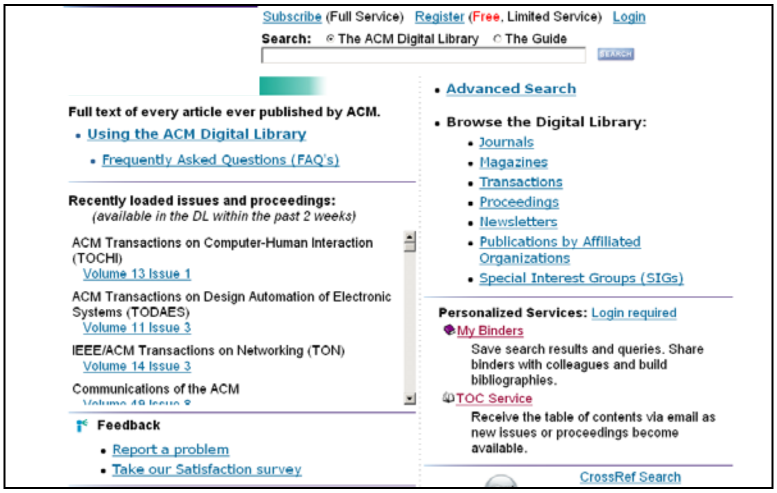
□ 2. Investigating actability dimensions: a language/action perspective on criteria for information systems evaluation
Agerfalk, Par J
Interacting with Computers; 16 (5) Oct 2004, pp.957-988
... primarily on business modelling and different business ***interaction patterns***. In this paper, nine dimensions of information systems from a LAP point of view are developed. The dimensions are founded on the notion that information systems used ...
View Record | S•F•X | InterLibrary Loan

From http://www.csa.com/csaillumina/login.php

How this pattern is related to others:

This pattern is related to **Result Records** and is a part of **Result Set**.

Information Retrieval system



Subscribe (Full Service) Register (Free, Limited Service) Login

Search: ● The ACM Digital Library ○ The Guide

Full text of every article ever published by ACM.

- Using the ACM Digital Library
  - Frequently Asked Questions (FAQ's)

Recently loaded issues and proceedings:
(available in the DL within the past 2 weeks)

ACM Transactions on Computer-Human Interaction (TOCHI)
Volume 13 Issue 1

ACM Transactions on Design Automation of Electronic Systems (TODAES)
Volume 11 Issue 3

IEEE/ACM Transactions on Networking (TON)
Volume 14 Issue 3

Communications of the ACM
Volume 49 Issue 8

Feedback

- Report a problem
- Take our Satisfaction survey

- Advanced Search
- **Browse the Digital Library:**
  - Journals
  - Magazines
  - Transactions
  - Proceedings
  - Newsletters
  - Publications by Affiliated Organizations
  - Special Interest Groups (SIGs)

Personalized Services: Login required

My Binders
Save search results and queries. Share binders with colleagues and build bibliographies.

TOC Service
Receive the table of contents via email as new issues or proceedings become available.

CrossRef Search

From http://portal.acm.org/dl.cfm

What:

An information retrieval system should allow users to perform general and advanced searches. An information retrieval system should allow users to browse the contents of the system, if appropriate. An information retrieval system should display the results of a query after the query has been executed. An information retrieval system should also provide help to those who may need assistance using the system.

Use When:

When users need to retrieve information that is stored in some type of system provide an information retrieval system that allows users to input queries and retrieve results. When the information is organized in some fashion that allows for browsing provide a browse function that lets the users look through the contents of the system.

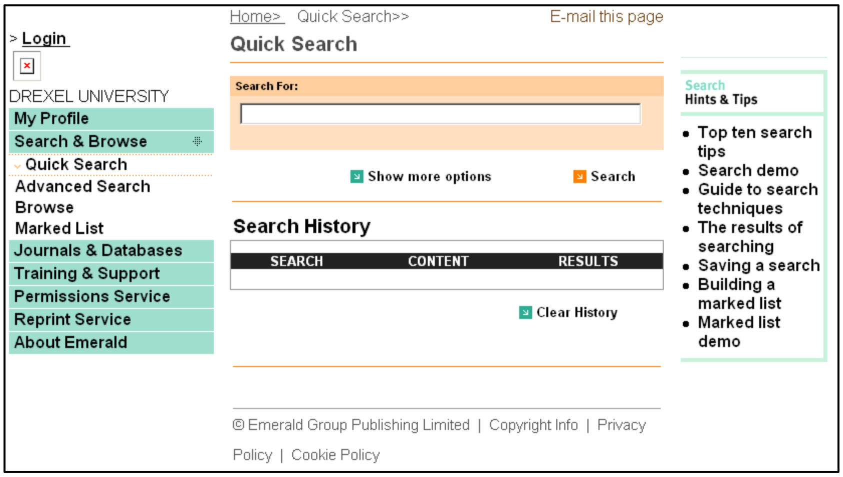
Why:

An Information Retrieval System should provide users with access to information through an interface that allows them to enter queries that express their information needs and be presented with the results of their queries. An information retrieval system should provide a general search and an advanced search because there will be users with different levels of experience. In addition users will have different information needs, therefore providing both a general and advanced search is appropriate. Sometimes a user may know exactly what they are looking for and other times they may have a general idea of what they are looking for, therefore an information retrieval system should provide a general search and an advanced search. In addition an information retrieval system should also have a browse function in order to allow users to browse the contents of the system for those that are not familiar with the types of information the system contains. An information retrieval system should provide a help function in order to help users use the different functions within the system.

How:

Provide both a general search and an advanced search interface. Be sure that the initial interface is a general search interface, but provide a link to the advanced search from the general search interface. Provide a link to the Help documentation on every system interface. Be sure that the Help is appropriate to what the user is trying to accomplish at every point in time. Provide a browse function on the initial search interface. After the user has input and executed a query display the results of the query.

Examples:

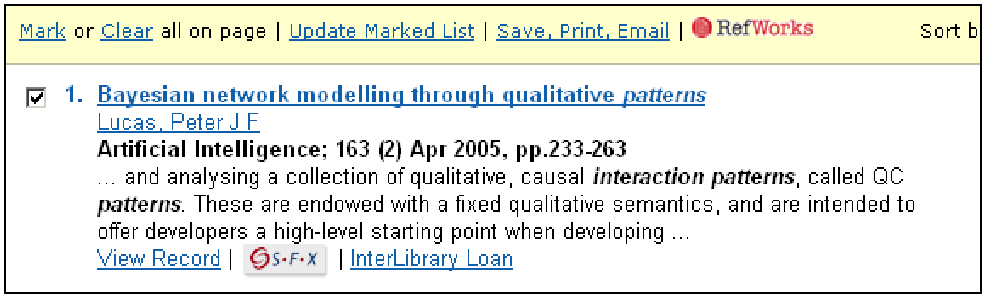


From http://www.emeraldinsight.com/Insight/

How this pattern is related to others:

This pattern contains **General Search**, **Advanced Search**, **Help**, **Preferences**, **Display Results**, and **Browse**.

Mark Results



Mark or Clear all on page | Update Marked List | Save, Print, Email | RefWorks Sort b

☑ 1. Bayesian network modelling through qualitative *patterns*
Lucas, Peter J F
Artificial Intelligence; 163 (2) Apr 2005, pp.233-263
... and analysing a collection of qualitative, causal ***interaction patterns***, called QC ***patterns***. These are endowed with a fixed qualitative semantics, and are intended to offer developers a high-level starting point when developing ...
View Record | S·F·X | InterLibrary Loan

From http://www.csa.com/csaillumina/login.php

What:

Allow users to mark results of interest and view, save or export the result records.

Use When:

When users are presented with the result set of a query they should have the option of marking result records they may want to save, export, or view later.

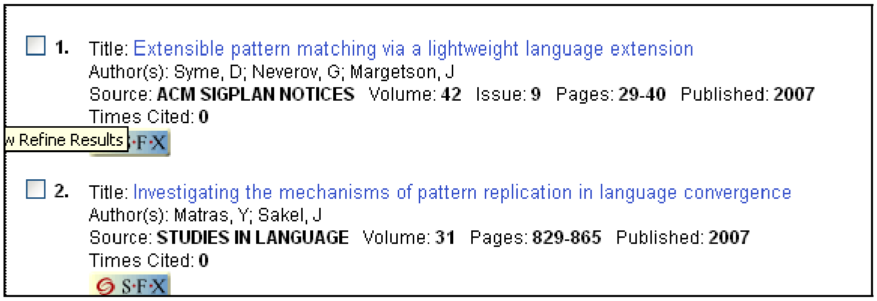
Why:

Users should be able to save or export citations or documents that meet their needs so that they may retrieve them again at a later time.

How:

Users should simply be able to check a box located to the left of the result records and then select from options for marking, saving, exporting, reviewing, etc. records or documents. Provide users with the option to Select all or Deselect all when marking documents.

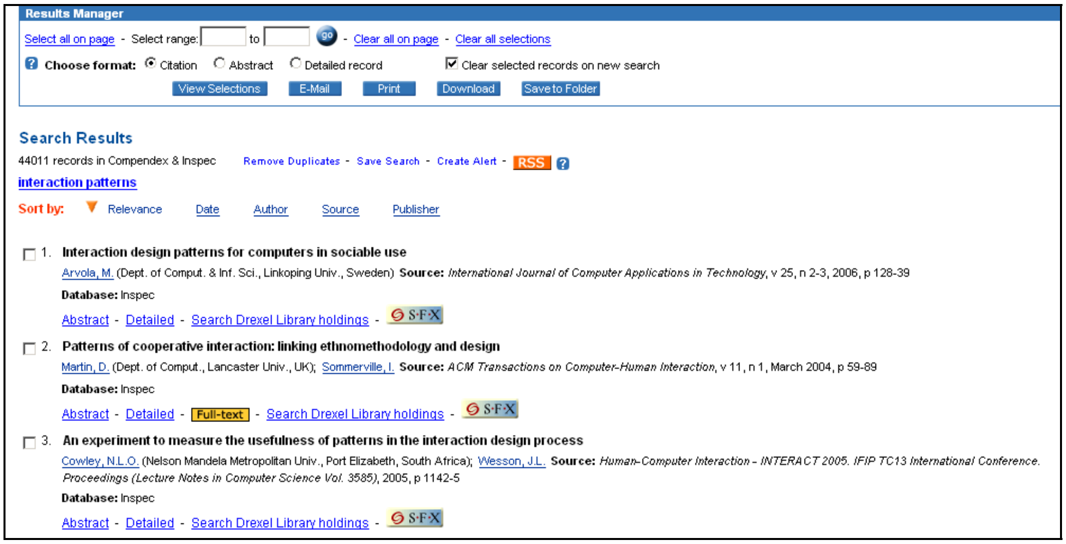
Examples:



□ **1.** Title: Extensible pattern matching via a lightweight language extension
Author(s): Syme, D; Neverov, G; Margetson, J
Source: **ACM SIGPLAN NOTICES** Volume: **42** Issue: **9** Pages: **29-40** Published: **2007**
Times Cited: **0**
w Refine Results ·F·X

□ **2.** Title: Investigating the mechanisms of pattern replication in language convergence
Author(s): Matras, Y; Sakel, J
Source: **STUDIES IN LANGUAGE** Volume: **31** Pages: **829-865** Published: **2007**
Times Cited: **0**
S·F·X

From http://apps.isiknowledge.com/


Results Manager
Select all on page - Select range: to go - Clear all on page - Clear all selections
Choose format: ● Citation ○ Abstract ○ Detailed record ☑ Clear selected records on new search
View Selections
E-Mail
Print
Download
Save to Folder

Search Results

44011 records in Compendex & Inspec

Remove Duplicates - Save Search - Create Alert - RSS
interaction patterns
Sort by: ▼ Relevance Date Author Source Publisher

□ 1. **Interaction design patterns for computers in sociable use**
Arvola, M. (Dept. of Comput. & Inf. Sci., Linkoping Univ., Sweden) **Source:** *International Journal of Computer Applications in Technology*, v 25, n 2-3, 2006, p 128-39
Database: Inspec
Abstract - Detailed - Search Drexel Library holdings - S·F·X

□ 2. **Patterns of cooperative interaction: linking ethnomethodology and design**
Martin, D. (Dept. of Comput., Lancaster Univ., UK); Sommerville, I. **Source:** *ACM Transactions on Computer-Human Interaction*, v 11, n 1, March 2004, p 59-89
Database: Inspec
Abstract - Detailed - Full-text - Search Drexel Library holdings - S·F·X

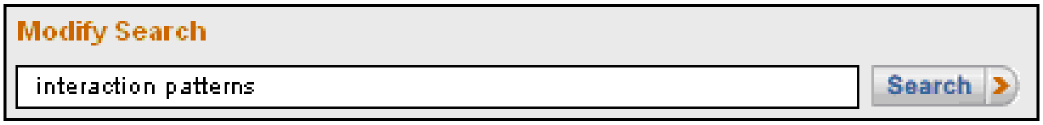
□ 3. **An experiment to measure the usefulness of patterns in the interaction design process**
Cowley, N.L.O. (Nelson Mandela Metropolitan Univ., Port Elizabeth, South Africa); Wesson, J.L. **Source:** *Human-Computer Interaction - INTERACT 2005. IFIP TC13 International Conference. Proceedings (Lecture Notes in Computer Science Vol. 3585)*, 2005, p 1142-5
Database: Inspec
Abstract - Detailed - Search Drexel Library holdings - S·F·X

From http://www.engineeringvillage2.org/

How this pattern is related to others:

This pattern is a part of **Display results**. This pattern is related to **Result Set**.

Modify Search



Modify Search

interaction patterns

From http://ieeexplore.ieee.org/Xplore/dynhome.jsp

What:

Allow users to change a search that has already been executed.

Use When:

After a user has already input and executed a search and received the results they should have this option.

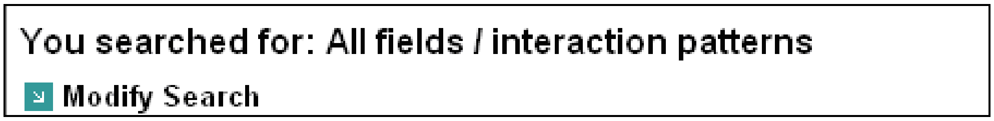
Why:

After a user has already executed a search they may realize that they want to revise their search. Providing users with the option of changing their current search allows more flexibility and saves time. Allowing users to change a search is easier and more convenient than forcing them to go back to the initial search interface to execute another query.

How:

Display the user’s current search query in a search box with a search button located next to the search box. Place the words “Modify Search” above the search box or place a link labeled modify search near the search box and button that displays the current search query.

Examples:



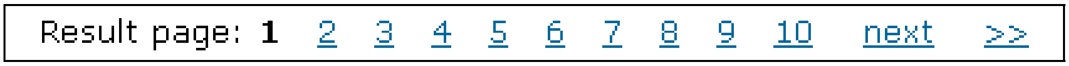
You searched for: All fields / interaction patterns

From http://www.emeraldinsight.com/Insight/

How this pattern is related to others:

This pattern is related to **General search**, **Advanced search**, **Display results** and **Executed Search Query displayed in Search box**.

Navigation of Results Pages



From http://portal.acm.org/dl.cfm

What:

The user should be able to navigate through the results pages by selecting Next, Previous, or the Results Page Number.

Use When:

When the user's query returns some results the user needs to be able to navigate through the results.

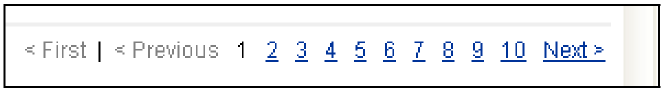
Why:

The user must be given a way to look through the results pages returned by a query. The user must be allowed to go from the first page of results to the next page of results and back to the previous page of results. In addition the user should be able to move to a particular page within the results by selecting a page number.

How:

The user should be given the option to go to the next page of results (if the user is not on the last page), the previous page of the results (if the user is not on the first page of results), or to select the results page that they would like to go to (unless there is only one page of results). There should be buttons for Next and Previous (when appropriate) and there should also be buttons that allow the user to skip to a particular page by selecting a number (1 2 3 4 5).

Examples:

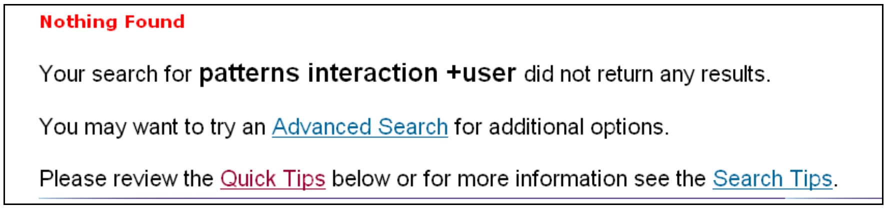


From http://proquest.umi.com/login

How this pattern is related to others:

This pattern is a part of **Display Results** and **Browse**. This pattern contains **Indicate Position in Results**.

No Results



Nothing Found

Your search for **patterns interaction +user** did not return any results.

You may want to try an Advanced Search for additional options.

Please review the Quick Tips below or for more information see the Search Tips.

From http://portal.acm.org/dl.cfm

What:

The user should be informed if their executed query returns no results.

Use When:

When a user's query does not return any results the user should be informed that there are no results which match their query.

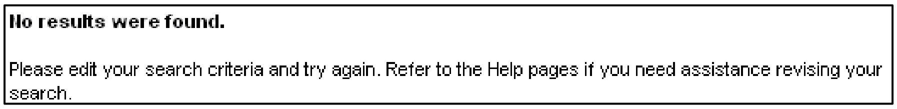
Why:

The user's goal is to retrieve information related to their search query. When the user's query returns no results they should be informed of this and directed to help documentation that may help them restructure their query. The user's search query should also be displayed so that they are aware of what their search query was.

How:

Display a message that informs the user that their query produced no results and provide a link to help documentation and search tips. The message could read "No results to display" or "Nothing found." Also, show the user what their search query was.

Examples:



No results were found.

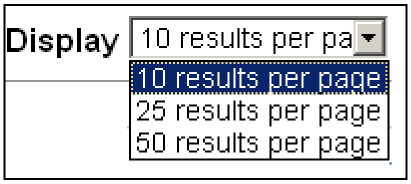
Please edit your search criteria and try again. Refer to the Help pages if you need assistance revising your search.

From http://ieeexplore.ieee.org/Xplore/dynhome.jsp

How this pattern is related to others:

This pattern is part of **IR systems**. This pattern is related to **Result Set**. This pattern is related to **Help**.

Number of Results to Display per Page



From http://portal.acm.org/dl.cfm

What:

Provide users with the option of changing the number of results displayed per page.

Use When:

When displaying the results of a user query allow for the flexibility to choose (from a limited number of options) the number of results to be displayed on each page.

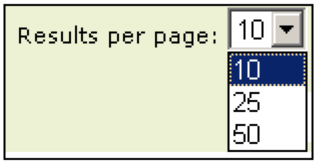
Why:

Users may want to view more or less results per page depending on their needs. Giving users the option to change the number of results per page allows them to customize the interface to meet their needs.

How:

Provide a pull-down that allows users to choose (from a limited number or options) the number of results to display per page.

Examples:

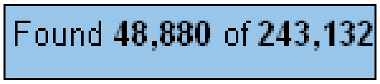


From http://www.csa.com/csaillumina/login.php

How this pattern is related to others:

This pattern is a part of **Display Results**. This pattern is related to **Results Set** and **Result Records**.

Number of retrieved results



Found **48,880** of **243,132**

From http://portal.acm.org/dl.cfm

What:

The total number of retrieved results should be given to the user.

Use When:

When the results of a user's query are displayed the user should be informed of the total number of results retrieved.

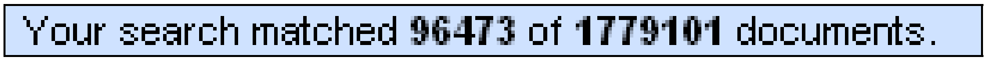
Why:

After executing a query the user should be able to judge whether or not to refine their search query. If a query returns very few results the user may need to expand their search or if a query returns too many results the user may want to refine their search.

How:

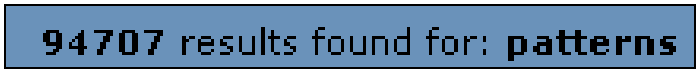
The total number of results retrieved for a query should be shown to the user. The total number of results should be displayed somewhere near the top portion of the interface near the *Search query displayed in the search box* and near the *Results navigation*.

Examples:



Your search matched **96473** of **1779101** documents.

From http://ieeexplore.ieee.org/Xplore/dynhome.jsp



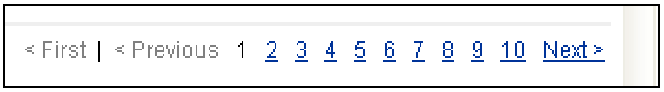
94707 results found for: **patterns**

From http://www.csa.com/csaillumina/login.php

How this pattern is related to others:

This pattern is a part of **Display Results**. This pattern is related to **Result Set**, **Result Records,** and **Position in Results**.

Position in Results



< First | < Previous 1 2 3 4 5 6 7 8 9 10 Next >

From http://proquest.umi.com/login

What:

The user's current position in the results pages should be indicated in the interface.

Use When:

When a user is looking through the results of a query they should be informed of what page they are currently viewing within the result set.

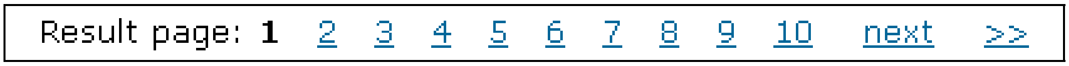
Why:

The system should always inform the user where they are within the result set so that they can be aware of what they have and have not looked at, for example if a user is on the 4th page of results and they have not found anything relevant it would be helpful if the system indicated that the user was on the 4th page of results so that the user could determine what to do next.

How:

When displaying the number of results pages the user's position should be indicated by differentiating the page number they are currently on from all the other results page numbers. This could be done by changing the color of the current page number, not underlining the current page number and underlining all the other page numbers, etc.

Examples:



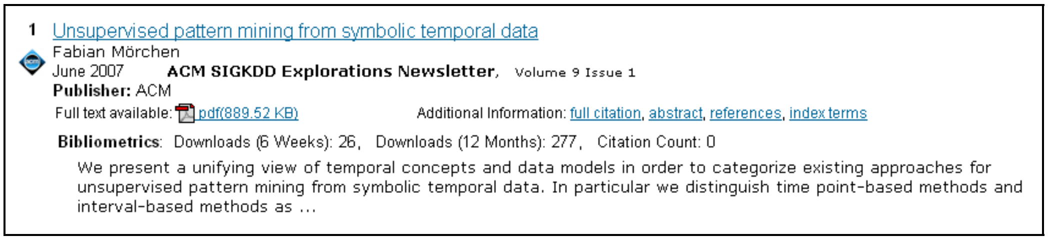
Result page: **1** 2 3 4 5 6 7 8 9 10 next >>

From http://portal.acm.org/dl.cfm

How this pattern is related to others:

This pattern is a part of **Results Navigation**. This pattern is related to **Result Set**, **Result Records,** and **Number of Retrieved Results**.

Result Record



1 Unsupervised pattern mining from symbolic temporal data
Fabian Mörchen
June 2007 **ACM SIGKDD Explorations Newsletter**, Volume 9 Issue 1
Publisher: ACM
Full text available: pdf(889.52 KB) Additional Information: full citation, abstract, references, index terms

Bibliometrics: Downloads (6 Weeks): 26, Downloads (12 Months): 277, Citation Count: 0

We present a unifying view of temporal concepts and data models in order to categorize existing approaches for unsupervised pattern mining from symbolic temporal data. In particular we distinguish time point-based methods and interval-based methods as ...

From http://portal.acm.org/dl.cfm

What:

The records that are contained in the result set returned by a user's query should be shown to the user. The result records should contain as much bibliographic information as possible. The user should be given the author(s), the title of the document, the source, publication year, volume, issue, etc. The user should also be given a link to the full text if available.

Use When:

When a user's executed query returns some results information about the individual results records should be presented to the user.

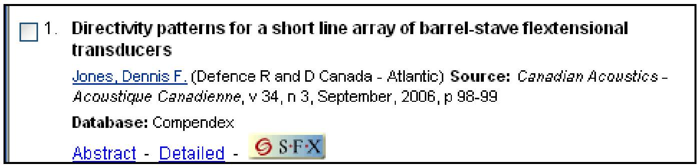
Why:

The user should be presented with the results records that match their executed query. The user should be given as much information as possible about the results records retrieved (for example: the author, title, source, publication year, etc) in order to determine if they have found what they are looking for.

How:

The user should be presented the title, the author(s), the source, the publication year, the publisher, the volume, issue, etc. The author and title should be presented first followed by the other bibliographic information.

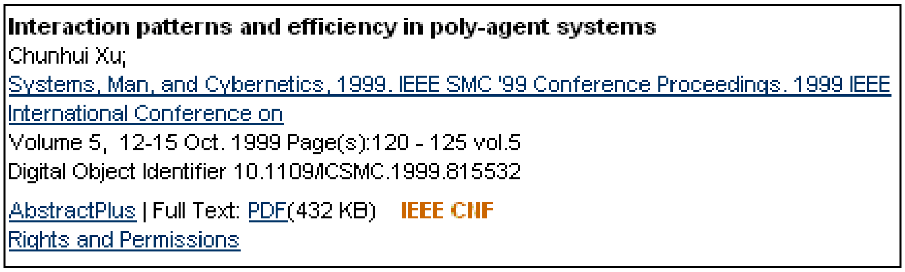
Examples:



□ 1. **Directivity patterns for a short line array of barrel-stave flextensional transducers**
Jones, Dennis F. (Defence R and D Canada - Atlantic) **Source:** *Canadian Acoustics - Acoustique Canadienne*, v 34, n 3, September, 2006, p 98-99
Database: Compendex
Abstract - Detailed - S·F·X

From http://www.engineeringvillage2.org/

Examples:



Interaction patterns and efficiency in poly-agent systems
Chunhui Xu;
Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on
Volume 5, 12-15 Oct. 1999 Page(s):120 - 125 vol.5
Digital Object Identifier 10.1109/ICSMC.1999.815532

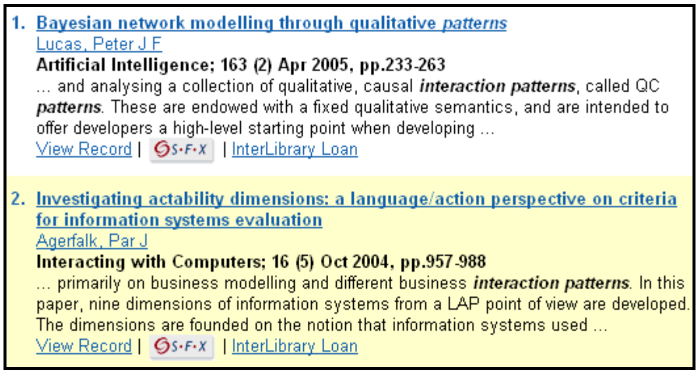
AbstractPlus | Full Text: PDF(432 KB) IEEE CNF
Rights and Permissions

From http://ieeexplore.ieee.org/Xplore/dynhome.jsp

How this pattern is related to others:

This pattern is part of **Result Set**. This pattern is related to **Display Format**, **Number of Retrieved Results**, **Position in Results** and **Indicate Search Terms in Result Set**.

Result Set



1. Bayesian network modelling through qualitative *patterns*
Lucas, Peter J F
Artificial Intelligence; 163 (2) Apr 2005, pp.233-263
... and analysing a collection of qualitative, causal ***interaction patterns***, called QC ***patterns***. These are endowed with a fixed qualitative semantics, and are intended to offer developers a high-level starting point when developing ...
View Record | S·F·X | InterLibrary Loan

2. Investigating actability dimensions: a language/action perspective on criteria for information systems evaluation
Agerfalk, Par J
Interacting with Computers; 16 (5) Oct 2004, pp.957-988
... primarily on business modelling and different business ***interaction patterns***. In this paper, nine dimensions of information systems from a LAP point of view are developed. The dimensions are founded on the notion that information systems used ...
View Record | S·F·X | InterLibrary Loan

From http://www.csa.com/csaillumina/login.php

What:

The result records of a query need to be presented to the user in an ordered, numbered list.

Use When:

After a user has executed a query all the results should be presented to the user unless there are no results in which case the user should be informed there are no results.

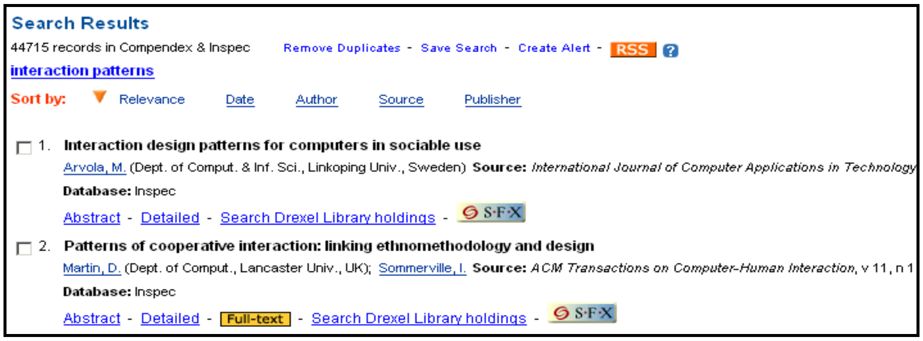
Why:

The user's goal is to find information therefore the system should present the user with the results it has found based on the user's query.

How:

The results of a user's query should be presented to the user in an ordered, numbered list.

Examples:



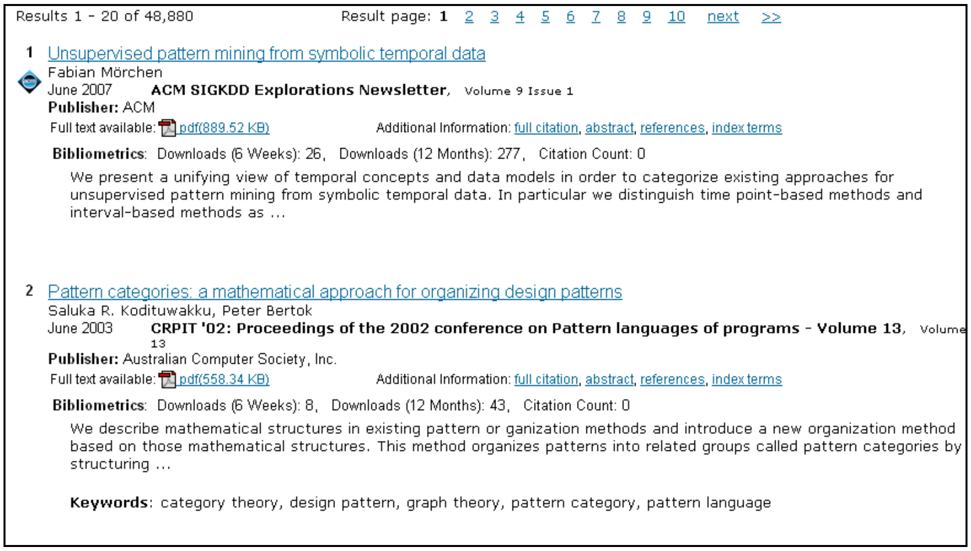
Search Results
44715 records in Compendex & Inspec   Remove Duplicates - Save Search - Create Alert - RSS ?
interaction patterns
Sort by: ▼ Relevance Date Author Source Publisher

□ 1. **Interaction design patterns for computers in sociable use**
Arvola, M. (Dept. of Comput. & Inf. Sci., Linkoping Univ., Sweden) **Source:** *International Journal of Computer Applications in Technology*
Database: Inspec
Abstract - Detailed - Search Drexel Library holdings - S·F·X

□ 2. **Patterns of cooperative interaction: linking ethnomethodology and design**
Martin, D. (Dept. of Comput., Lancaster Univ., UK); Sommerville, I. **Source:** *ACM Transactions on Computer-Human Interaction*, v 11, n 1
Database: Inspec
Abstract - Detailed - Full-text - Search Drexel Library holdings - S·F·X

From http://www.engineeringvillage2.org/

Examples:



Results 1 - 20 of 48,880 Result page: **1** 2 3 4 5 6 7 8 9 10 next >>

1 Unsupervised pattern mining from symbolic temporal data
Fabian Mörchen
June 2007 **ACM SIGKDD Explorations Newsletter**, Volume 9 Issue 1
Publisher: ACM
Full text available: pdf(889.52 KB) Additional Information: full citation, abstract, references, index terms

Bibliometrics: Downloads (6 Weeks): 26, Downloads (12 Months): 277, Citation Count: 0

We present a unifying view of temporal concepts and data models in order to categorize existing approaches for unsupervised pattern mining from symbolic temporal data. In particular we distinguish time point-based methods and interval-based methods as ...

2 Pattern categories: a mathematical approach for organizing design patterns
Saluka R. Kodituwakku, Peter Bertok
June 2003 **CRPIT '02: Proceedings of the 2002 conference on Pattern languages of programs - Volume 13**, Volume 13
Publisher: Australian Computer Society, Inc.
Full text available: pdf(558.34 KB) Additional Information: full citation, abstract, references, index terms

Bibliometrics: Downloads (6 Weeks): 8, Downloads (12 Months): 43, Citation Count: 0

We describe mathematical structures in existing pattern or ganization methods and introduce a new organization method based on those mathematical structures. This method organizes patterns into related groups called pattern categories by structuring ...

Keywords: category theory, design pattern, graph theory, pattern category, pattern language

From http://portal.acm.org/dl.cfm

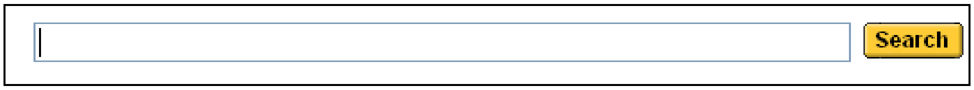
How this pattern is related to others:

This pattern is a part of **Display Results**.

This pattern contains **Result Records** and **Indicate search terms in result set**.

This pattern is related to **No Results**. This pattern is also related to **Search History**, **Sort By**, **Sorting Order***r*, **Display Format** and **Mark Documents**.

Search Box and Button



http://proquest.umi.com/

What:

The search interface must contain a search box in which the user can input their query or part of their query. The search interface must also contain a button that reads SEARCH in which the user can press to execute the search.

Use When:

Whenever a user may need to enter a query or part of a query this is appropriate.

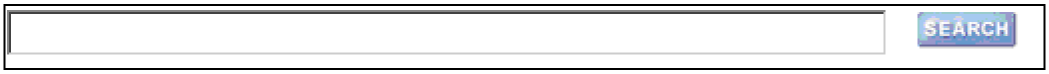
Why:

The user must have a designated space to enter their search query or part of their search query. The system must have a source for obtaining input.

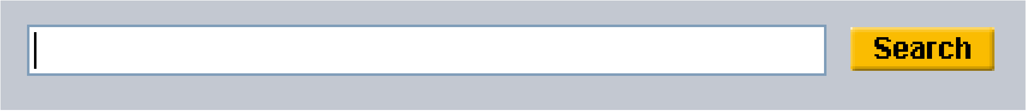
How:

Provide a search box with a search button located to the right of the search box.

Examples:



From http://portal.acm.org/dl.cfm



From http://www.engineeringvillage2.org/

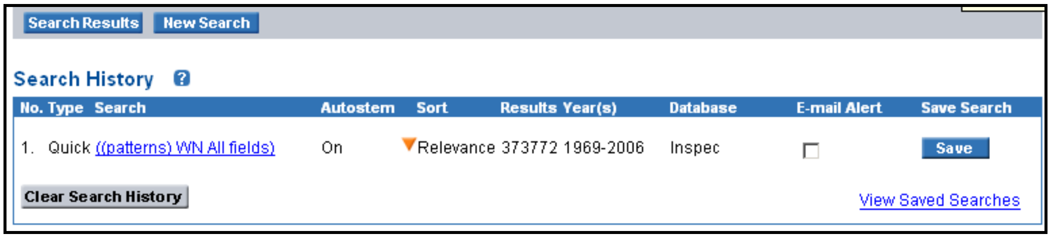
How this pattern is related to others:

This pattern is part of **General search** and **Advanced search**.

This pattern is related to **Executed Search Query Displayed in Search Box**.

This pattern contains **Execute search on ENTER** and **Execute search by clicking SEARCH**.

Search History



Search History

No.	Type	Search	Autostem	Sort	Results	Year(s)	Database	E-mail Alert	Save Search
1.	Quick	((patterns) WN All fields)	On	Relevance	373772	1969-2006	Inspec	□	Save

From http://www.engineeringvillage2.org/

What:

Provide users with a history of the search queries they have executed and a summary of the results.

Use When:

When a user has performed one or more queries they should have the option to view their past queries and the results of those queries.

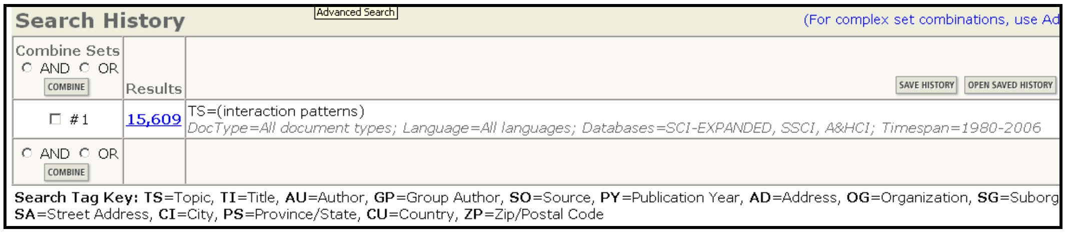
Why:

Users may want to save and or compare the results of their previous search queries. User may also want to view all the queries they have already performed. This may be used when users may be performing complex queries that they may want to refine and/or combine with previous results sets.

How:

Provide users with a history of all their queries, include what the search terms were and what the results were.

Examples:



Search History

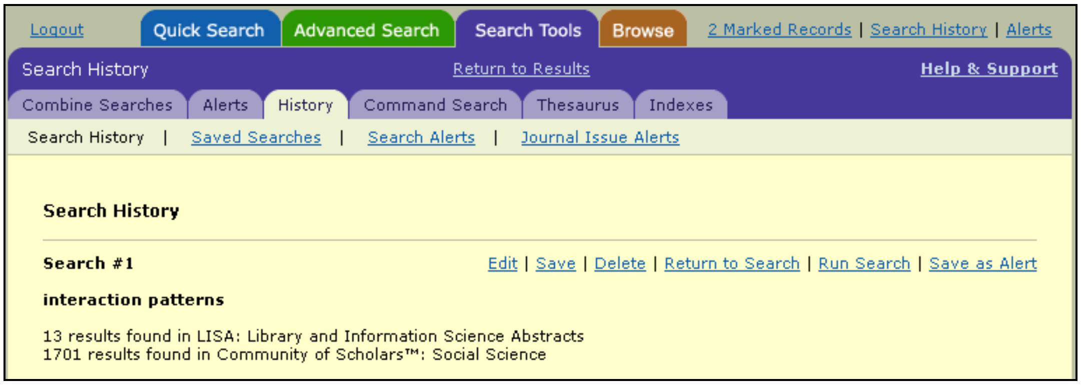
(For complex set combinations, use Ad

Combine Sets ○ AND ○ OR	Results	
□ #1	15,609	TS=(interaction patterns) *DocType=All document types; Language=All languages; Databases=SCI-EXPANDED, SSCI, A&HCI; Timespan=1980-2006*
○ AND ○ OR		

Search Tag Key: TS=Topic, **TI**=Title, **AU**=Author, **GP**=Group Author, **SO**=Source, **PY**=Publication Year, **AD**=Address, **OG**=Organization, **SG**=Suborg
SA=Street Address, **CI**=City, **PS**=Province/State, **CU**=Country, **ZP**=Zip/Postal Code

From http://apps.isiknowledge.com/

Examples:


Logout
Quick Search
Advanced Search
Search Tools
Browse
2 Marked Records | Search History | Alerts
Search History
Return to Results
Help & Support
Combine Searches
Alerts
History
Command Search
Thesaurus
Indexes
Search History | Saved Searches | Search Alerts | Journal Issue Alerts
Edit | Save | Delete | Return to Search | Run Search | Save as Alert

Search History

Search #1

interaction patterns

13 results found in LISA: Library and Information Science Abstracts
1701 results found in Community of Scholars™: Social Science

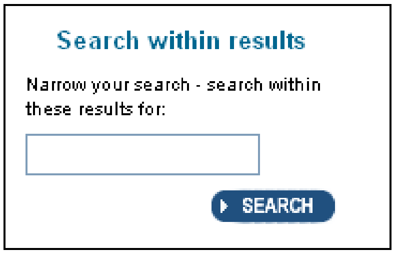
From http://www.csa.com/csaillumina/login.php

How this pattern is related to others:

This pattern is a part of **Display results**.

This pattern is related to **Result Set** and **Number of Retrieved Results**.

Search within results



Search within results

Narrow your search - search within these results for:

From http://www.morganclaypool.com/

What:

Provide users with the option to search within the results of a previous query.

Use When:

When the user has already entered a query and received the results of the query a user may want to narrow their search therefore provide the option to search within the results.

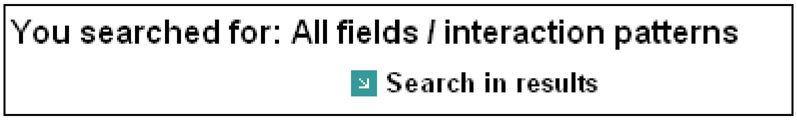
Why:

Users may want to search further within the results of a particular query if the result set returned was large or they feel they would like to refine their search to find something within the result set.

How:

Provide an option to search within the results by providing a link to a new interface that contains a search box and button labeled “search within results” or provide a search box and button labeled “search within results” on the interface that displays the results.

Examples:



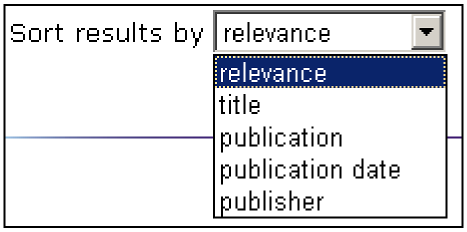
You searched for: All fields / interaction patterns

From http://www.emeraldinsight.com/Insight/

How this pattern is related to others:

This pattern is related to **General Search**, **Advanced Search**, **Display Results**, and **Executed Search Query displayed in search box**.

Sort results by


Sort results by relevance

From http://portal.acm.org/dl.cfm

What:

Provide multiple options for sorting the results of a query.

Use When:

When returning the results of a user's query the results need to be sorted in some manner and the user needs to be made aware of how the results are being sorted. If there are many appropriate ways that the result set can be displayed provide a default way of presenting the results and allow the users to select other ways of organizing the results both before and after they have entered a query.

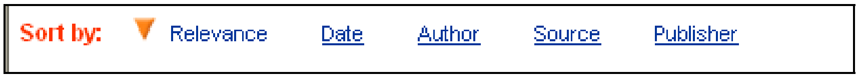
Why:

There must be a default way of presenting and organizing the results (possibly by relevance) but sometimes the default way of presenting the results may not be what the user prefers in a particular situation. Therefore different options for organizing the results should be available to the user both before and after executing a query in order to allow for greater flexibility.

How:

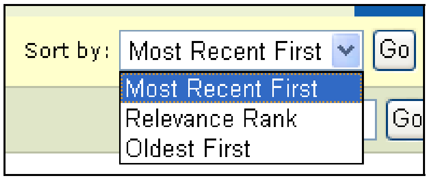
The system should sort the results based on a default strategy (possibly by relevance) and also provide a number of other options (that the user can select either from a dropdown or using radio buttons) for sorting the results both before and after a query has been entered. Depending on the users information need they may find it more appropriate to view the results by relevance, publication date, journal, etc.

Examples:



From http://www.engineeringvillage2.org/

Examples:



Sort by: Most Recent First

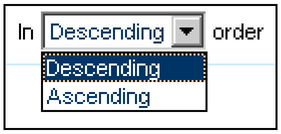
From http://www.csa.com/csaillumina/login.php

How this pattern is related to others

This pattern is a part of **Display results**. This pattern is related to **Result set**.

The IR system needs to return results in some organized fashion, but the chosen fashion may not be want the user prefers for a particular situation, therefore there should be options for **displaying results** that can be easily changed before a search and after a search is executed. Users should also be able to further specify the order the results are shown in, for example **sorting order**.

Sorting Order



From http://portal.acm.org/dl.cfm

What:

Provide users with the option to sort records in ascending and descending order.

Use When:

When the results of a query are presented to a user they are organized in some fashion. Allow the users to manipulate the order of the records.

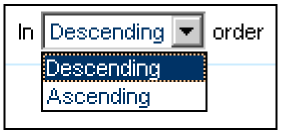
Why:

Allow users the flexibility to sort records in both ascending and descending order so that they are not restricted to only viewing records in ascending or descending order.

How:

Provide the option to sort results in both ascending and descending order.

Examples:

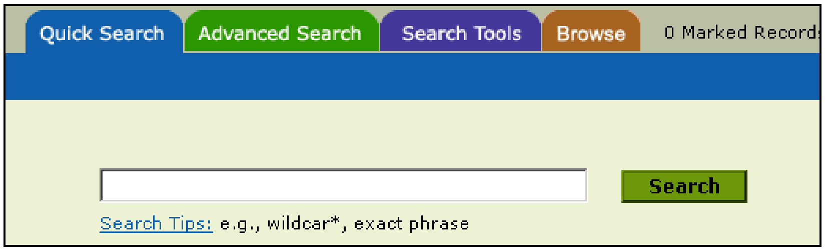


From http://portal.acm.org/dl.cfm

How this pattern is related to others:

This pattern is a part of **Display Results**. This pattern is related to **Result set and sort results by**.

Tab Navigation



Search Tips: e.g., wildcar*, exact phrase

From http://www.csa.com/csaillumina/login.php

What:

Provide tabs that allow users to navigate to the different functions in the information retrieval system. Include a tab for general search, advanced search, browse, help and possibly search history.

Use When:

When an information retrieval system provides multiple functions be sure to allow users to easily view what they can do with the system and also easily navigate from one part of the system to another.

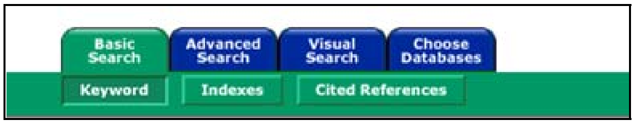
Why:

Providing navigation that is always present in the interface allows users to move to any section of the system they choose at any point in time. The navigation structure also communicates to the users to what the system can do for them without forcing them to explore to find out. Tabs allow for presenting all the systems options to the user at all times.

How:

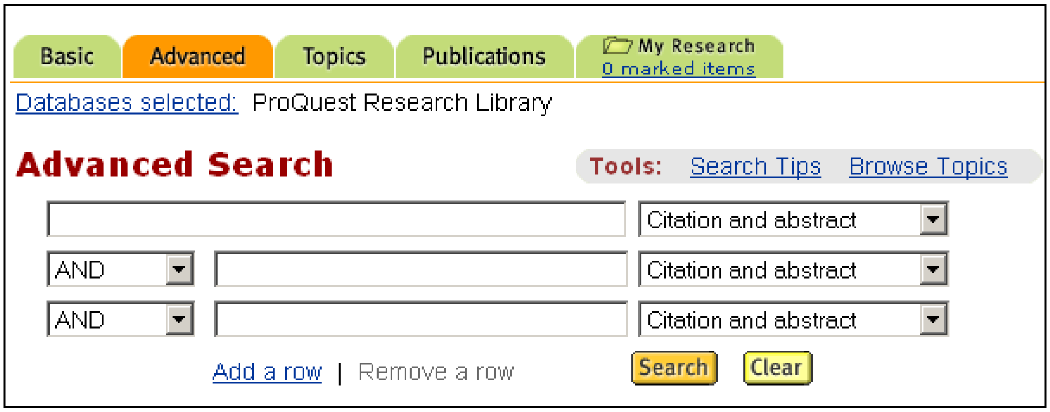
Provide Tabs on the top portion of the interface that are color coded based on the functionality contained within them. When a user selects a tab the user should be presented with the appropriate options and the tab should be “selected” in the interface and differentiated from all the other tabs.

Examples:



From http://www.ebscohost.com/

Examples:



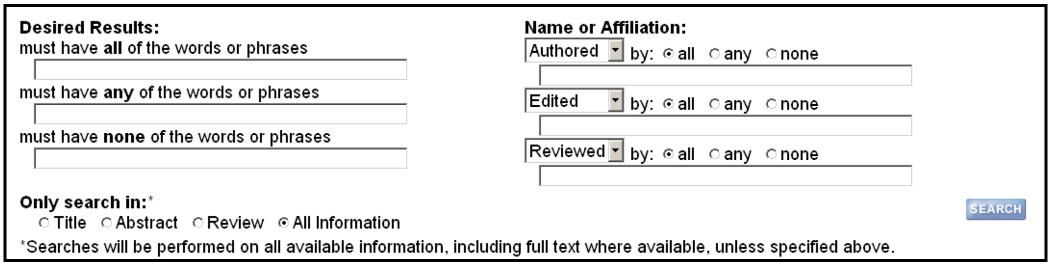
Databases selected: ProQuest Research Library

From http://proquest.umi.com/

How this pattern is related to others:

This pattern is a part of **Information Retrieval Systems**. This pattern is related to **General Search**, **Advanced Search**, **Browse**, **Help** and **Search History**.

Tab through Fields



Desired Results:
must have **all** of the words or phrases
must have **any** of the words or phrases
must have **none** of the words or phrases

Name or Affiliation:
Authored by: ● all ○ any ○ none
Edited by: ● all ○ any ○ none
Reviewed by: ● all ○ any ○ none

Only search in:*
○ Title ○ Abstract ○ Review ● All Information
*Searches will be performed on all available information, including full text where available, unless specified above.

From http://portal.acm.org/dl.cfm

What:

Allow users to navigate from one advanced search field to another by pressing tab.

Use When:

Use this pattern when a user is presented with more than one search field that could be filled in. This should be used in cases where there are multiple fields in which a user can enter information, for example in an advanced search interface.

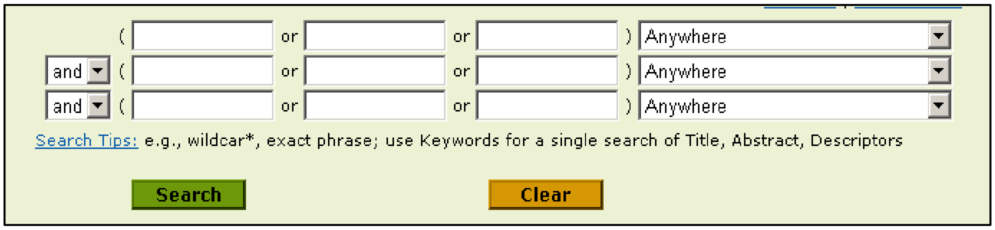
Why:

When there are multiple fields that a user may want to enter search terms using the tab to move from field to field is much faster that forcing the users to move the mouse to every search field.

How:

Allow users to navigate from one field to the other by pressing tab. Also be sure that the tab order is in logical order, for example tab from the first field to the second field, then to the third field, not from the first field to the third field and then back to the second field.

Examples:

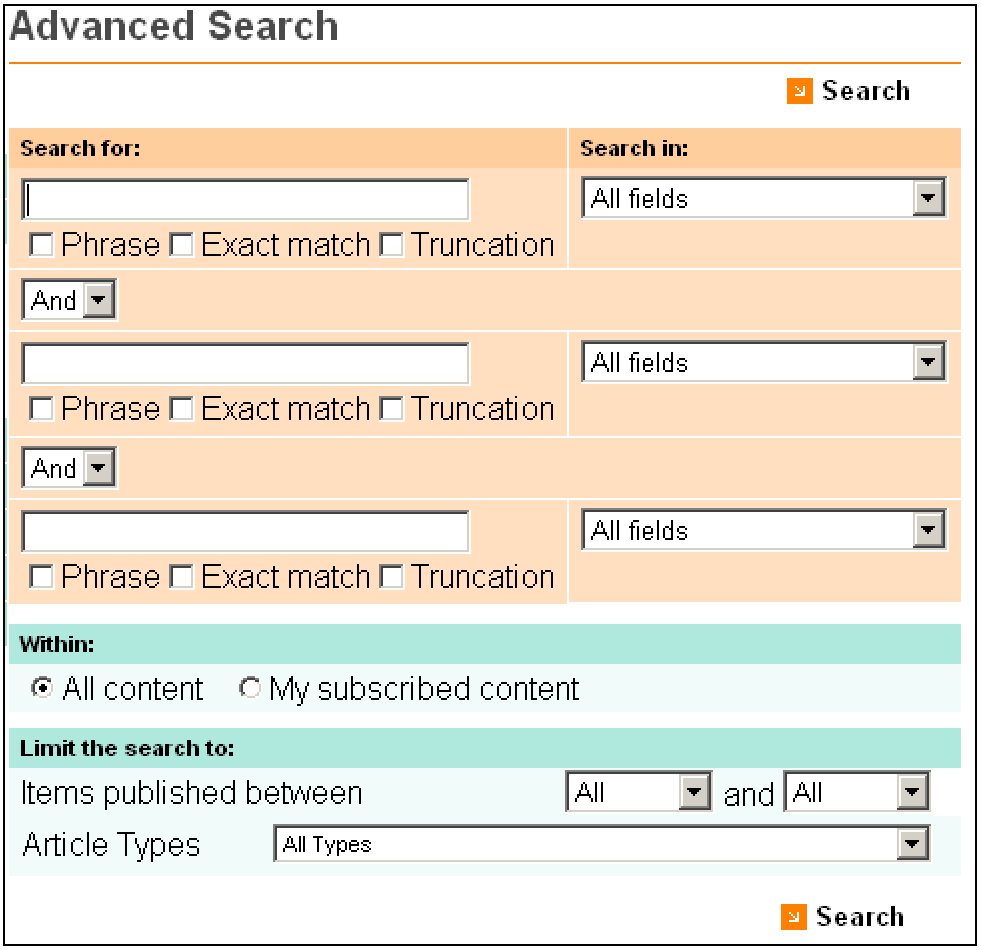


(or or) Anywhere
and (or or) Anywhere
and (or or) Anywhere

Search Tips: e.g., wildcar*, exact phrase; use Keywords for a single search of Title, Abstract, Descriptors

From http://www.csa.com/csaillumina/login.php

Examples:



Advanced Search

Search

Search for:	Search in:
☐ Phrase ☐ Exact match ☐ Truncation	All fields
And	
☐ Phrase ☐ Exact match ☐ Truncation	All fields
And	
☐ Phrase ☐ Exact match ☐ Truncation	All fields

Within:

● All content ○ My subscribed content

Limit the search to:

Items published between All and All

Article Types All Types

Search

From http://www.emeraldinsight.com/Insight/

How this pattern is related to others:

This pattern is a part of **Advanced search fields** and **Grouping Advanced Search fields**.

Appendix F. Pattern Sorting Instructions

We are trying to investigate how people view the relationships between these patterns

.

Therefore we are asking you to please sort the following 39 patterns into piles of related patterns.

Keep in mind there is no right or wrong way of doing this. You may create as many piles as you see fit and you may place as many patterns as you would like in a pile.

After you have sorted the patterns or while you are sorting the patterns please name each resulting pile using one of these post-it notes. The name may be as long of as short as you would like. If you are not able to name the group of patterns for some reason please still place a post-it note on the group.

Again, remember there is no right or wrong way to do this. We do not fully understand how these patterns are related and this is why we are asking a number of people to sort the patterns into piles of related patterns.

Appendix G. Guidelines Overview/Tutorial

Overview of Guidelines in HCI

Guidelines can be used to design systems in HCI. A guideline is a recommendation of good practice. The primary goal of a guideline is to improve the consistency and quality of the user interface. Guidelines help communicate best practices for user interfaces.

Each guideline contains the following elements:

- a name
- a description

Appendix H. Interface Template

Please sketch your information retrieval system interfaces in the boxes below.

Appendix I. Post Task Questionnaires

Patterns Group Questionnaire

Thank you for your participation.

Please place an X in the box that expresses your level of agreement or disagreement with the following statements.

	Strongly Disagree	Disagree	Neither agree nor disagree	Agree	Strongly Agree
I understood the patterns.					
I think the patterns helped me design the interfaces.					
I clearly understood how to use the patterns.					
I would use the patterns in the future.					
I think all designers should use patterns.					
I understood what I was asked to do.					
I enjoyed this activity.					

What did you like about this activity?

____

____

____

What did you dislike about this activity?

____

____

____

Guidelines Group Questionnaire

Thank you for your participation.

Please place an X in the box that expresses your level of agreement or disagreement with the following statements.

	Strongly Disagree	Disagree	Neither agree nor disagree	Agree	Strongly Agree
I understood the guidelines.					
I think the guidelines helped me design the interfaces.					
I clearly understood how to use the guidelines.					
I would use the guidelines in the future.					
I think all designers should use guidelines.					
I understood what I was asked to do.					
I enjoyed this activity.					

Please answer the following questions.

What did you like about this activity?

____

____

____

What did you dislike about this activity?

____

____

____

Control Group Questionnaire

Thank you for your participation.

Please place an X in the box that expresses your level of agreement or disagreement with the following statements.

	Strongly Disagree	Disagree	Neither agree nor disagree	Agree	Strongly Agree
I understood what I was asked to do.					
I enjoyed this activity.					

Please answer the following questions.

What did you like about this activity?

____

____

____

What did you dislike about this activity?

____

____

____

Appendix J. Evaluation Form for Participants' Interfaces

Please use this form to rate the quality of the interface designs. Please keep in mind that you are not judging the graphic design of the interfaces or neatness of the interfaces but rather you are judging the overall functionality of the interfaces as expressed in the design task.

Interface Number ______________

Please use the following scale to rate the quality of information retrieval interfaces. Please rate each of the quality elements based on the definitions below.

Ease of Use – How easy is it to use the system interfaces?

Level of Detail – How low-level, readily implementable, and non-vague are the elements of the design?

Completeness – How complete is the design? Does the design contain all the necessary parts that it will need to work?

Overall Quality – Overall how good of a solution is the design?

Ease of use	**Not at all** **Easy to use**						**Very easy** **to use**
	1	2	3	4	5	6	7
Level of Detail	**Not at all** **Detailed**						**Very** **Detailed**
	1	2	3	4	5	6	7
Completeness	**Not at all** **Complete**						**Very** **Complete**
	1	2	3	4	5	6	7
Overall Quality	**Very Poor** **Quality**						**Excellent** **Quality**
	1	2	3	4	5	6	7

Comments:

____

____

Appendix K. Expanded Definitions of Quality Elements for Evaluating Interfaces

<u>Ease of Use</u> – How easy is it to use the system interfaces?

- Consistency
- Navigation
    - From one interface to the next
    - To more results pages
- Options (to customize interfaces)
    - number of results to display
    - sort results by
- Ability to change query from results interface
- Highlighting search terms in the results

<u>Level of Detail</u> – How low-level, readily implementable, and non-vague are the elements of the design?

<u>Completeness</u> – How complete is the design? Does the design contain all the necessary parts that it will need to work?

Does the design contain all the interfaces described in the design task?
(The interfaces do not necessarily need to be in the correct order)

- General Search
- Results
- Help
- Advanced Search
- Results

Is it possible to get from one interface to the next? (navigation)

<u>Overall Quality</u> – Overall how good of a solution is the design?

Appendix L. List of IR Systems examined in recording patterns

ACM Digital Library

http://portal.acm.org/dl.cfm?coll=portal&dl=ACM&CFID=2519336&CFTOKEN=46047674

American history and life

http://serials.abc-clio.com/active/start?_appname=serials&initialdb=AHL

Blackwell synergy http://www.blackwell-synergy.com/

ChemVillage

http://www.chemvillage.org/c/s/C?EISESSION=1_eafb7110dd24970e731a0138128814&CID=quickSearch&database=384

ComAbsracts http://www.cios.org/www/absrch.htm

CSA http://www-md1.csa.com/ids70/quick_search.php?SID=4fh7bnnilhbj9ntmub351etmf6

CQ Researcher http://library.cqpress.com/cqresearcher/

Digital Dissertations http://wwwlib.umi.com/dissertations/search

Dynamed http://dynamed102.epnet.com/AlphaBrowse.aspx?Letter=A

Ebrary

http://site.ebrary.com/lib/drexel/Top?layout=search&nosr=1&p00=&f00=text&p01=&f01=subject&d=all&l=all&frm=adv.x&smp.x=37&smp.y=6

EbscoHost

http://web.ebscohost.com/ehost/search?hid=117&sid=9850bf8a-3232-403a-b28f-b120617b9dec%40sessionmgr102

Eighteenth Century Collections Online

http://galenet.galegroup.com/servlet/ECCO;jsessionid=4144574FA12A8491A18985389DE0CDC2?locID=drexel_law

Engineering village

http://www.engineeringvillage2.org/controller/servlet/Controller?CID=quickSearch&database=INSPEC

Emerald

http://www.emeraldinsight.com/Insight/menuNavigation.do?hdAction=menu_ca_sb_quick_search

ERIC http://www.eric.ed.gov/ERICWebPortal/Home.portal

Expanded Academic

http://find.galegroup.com/itx/start.do?prodId=EAIM&userGroupName=drexel_main

Factiva http://global.factiva.com/sb/default.aspx?NAPC=S&fcpil=en

FSTA Direct http://www.fstadirect.com/AdvancedsearchPage.asp

Gale

http://find.galegroup.com/gvrl/start.do?prodId=GVRL&userGroupName=drexel_law&finalAuth=true

IEEE http://ieeexplore.ieee.org/Xplore/dynhome.jsp

JSTOR http://www.jstor.org/search/

MathSciNet http://www.ams.org/mathscinet/search.html

NetLibrary http://www.netlibrary.com/library_home_page.asp

OCLC

http://newfirstsearch.oclc.org/WebZ/FSPrefs?entityjsdetect=:javascript=true:screensize=large:sessionid=fsapp6-59010-esxgxk54-35u8ci:entitypagenum=1:0

Proquest

http://proquest.umi.com/pqdweb?RQT=302&COPT=U0ZEPTImU01EPTYmSU5UPTAmVkVSPTImREJTPUc2&clientId=18133&cfc=1

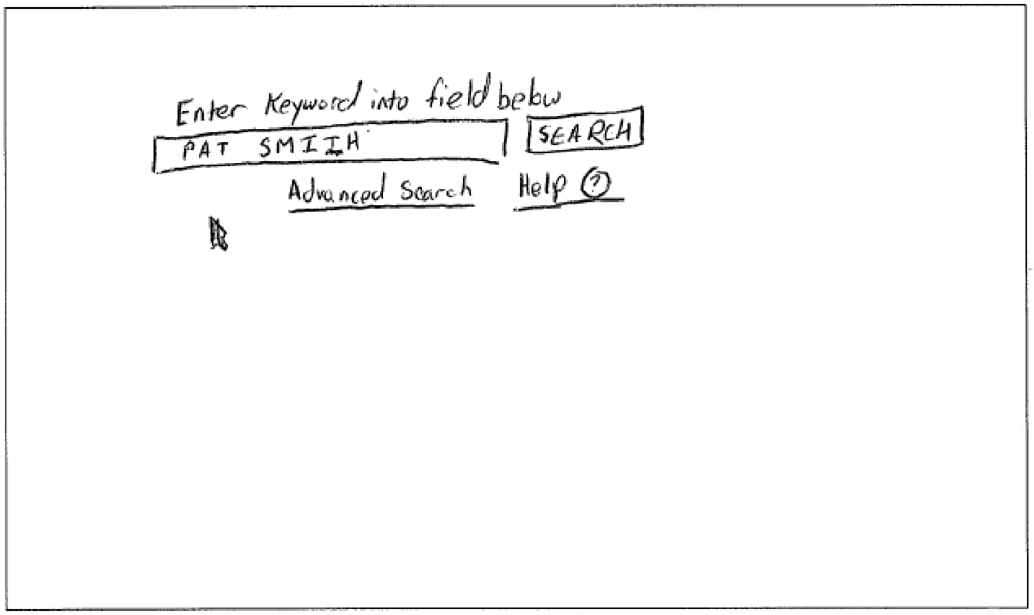
Science direct http://www.sciencedirect.com/

Web of science http://portal.isiknowledge.com/portal.cgi?DestApp=WOS&Func=Frame

Wiley http://www3.interscience.wiley.com/search/allsearch

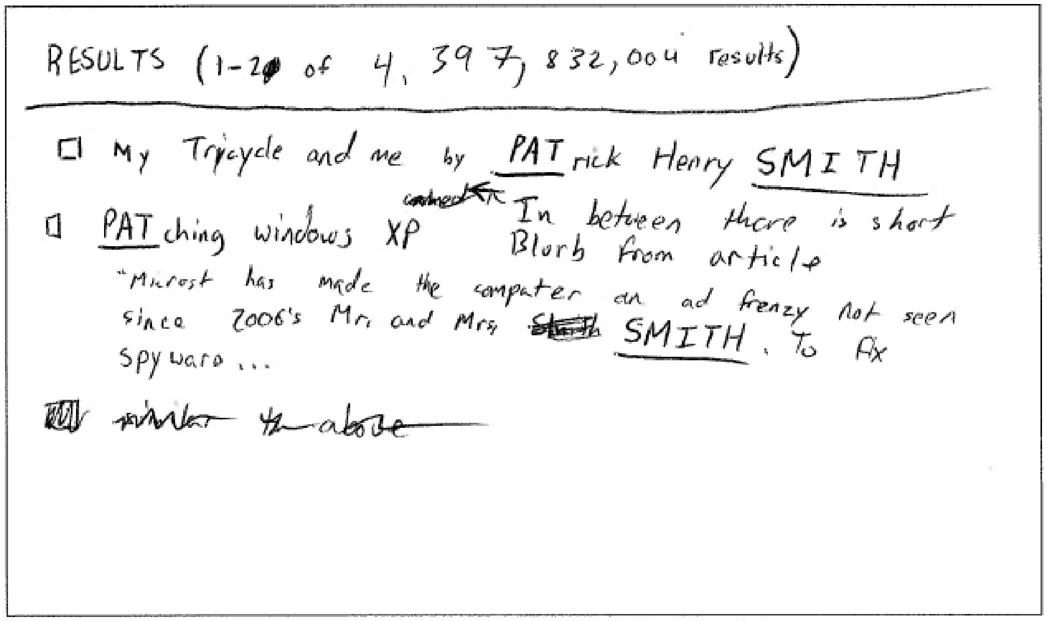
Wilson web http://vnweb.hwwilsonweb.com/hww/shared/shared_main.jhtml?_requestid=227069

Appendix M. Examples of Participants Designs



Enter Keyword into field below

PAT SMITH



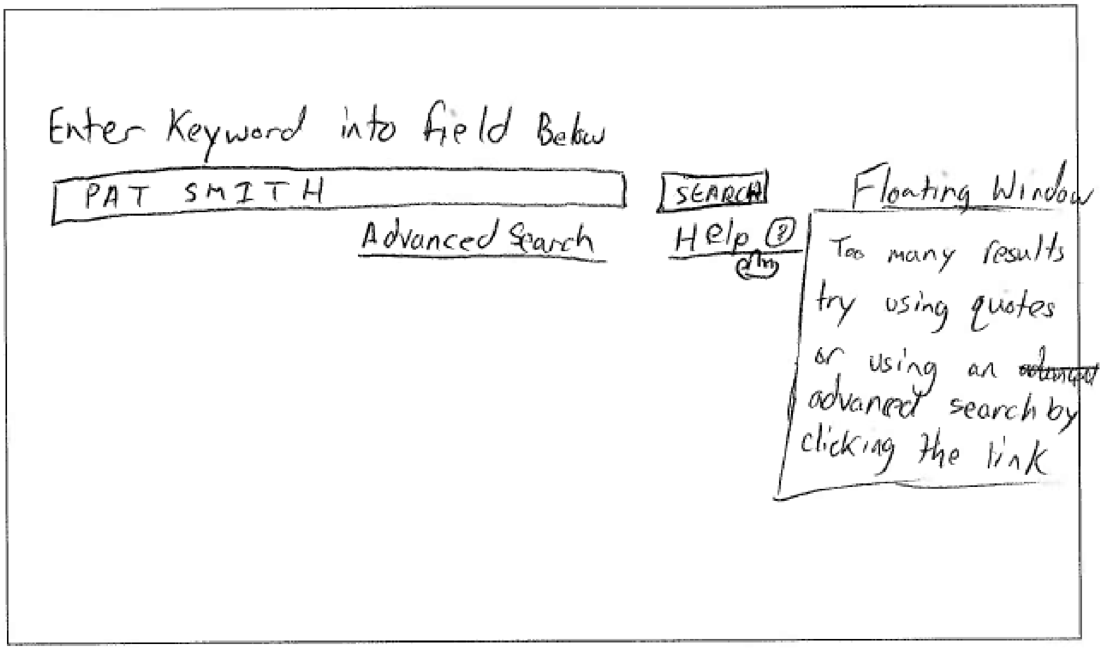
RESULTS (1-20 of 4,397,832,004 results)

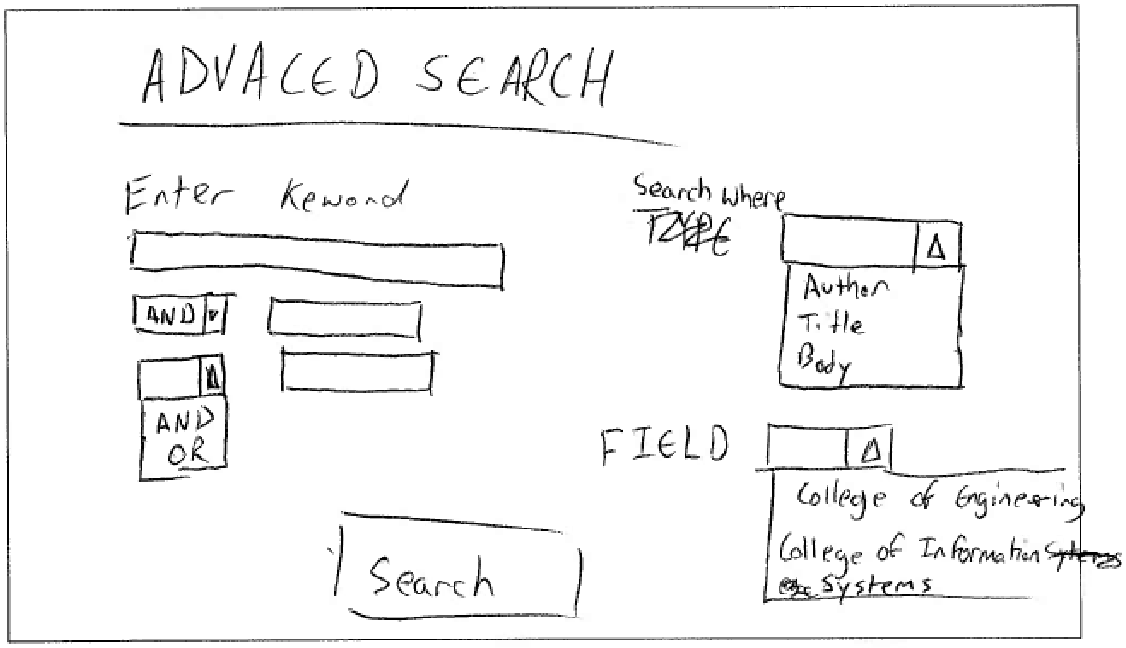
□ My Tricycle and me by PATrick Henry SMITH

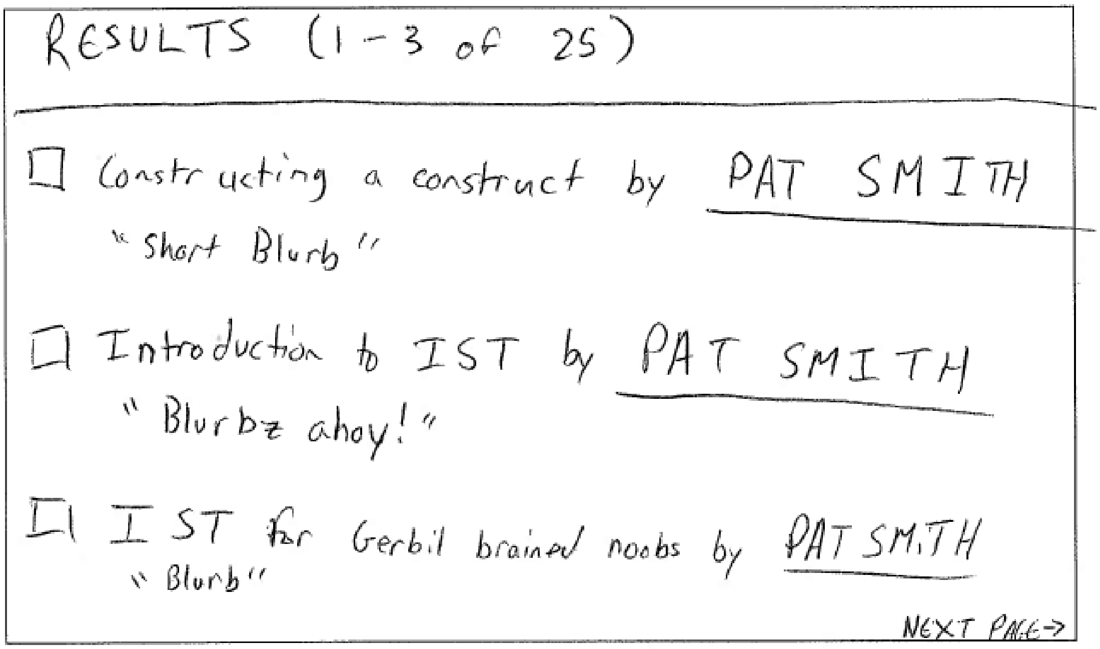
□ PATching windows XP content In between there is short Blorb from article

"Microst has made the computer an ad frenzy not seen since 2006's Mr. and Mrs. ~~Smith~~ SMITH. To fix spyware . . .

☒ ~~similar to above~~


Enter Keyword into field Below
PAT SMITH
SEARCH
Advanced Search
Help ?
Floating Window
Too many results try using quotes or using an ~~advanced~~ advanced search by clicking the link


ADVACED SEARCH
Enter Keword
AND
AND
OR
Search where
~~TYPE~~
Author
Title
Body
FIELD
College of Engineering
College of Information ~~Systems~~ Systems
Search



RESULTS (1 – 3 of 25)

☐ Constructing a construct by PAT SMITH

"Short Blurb"

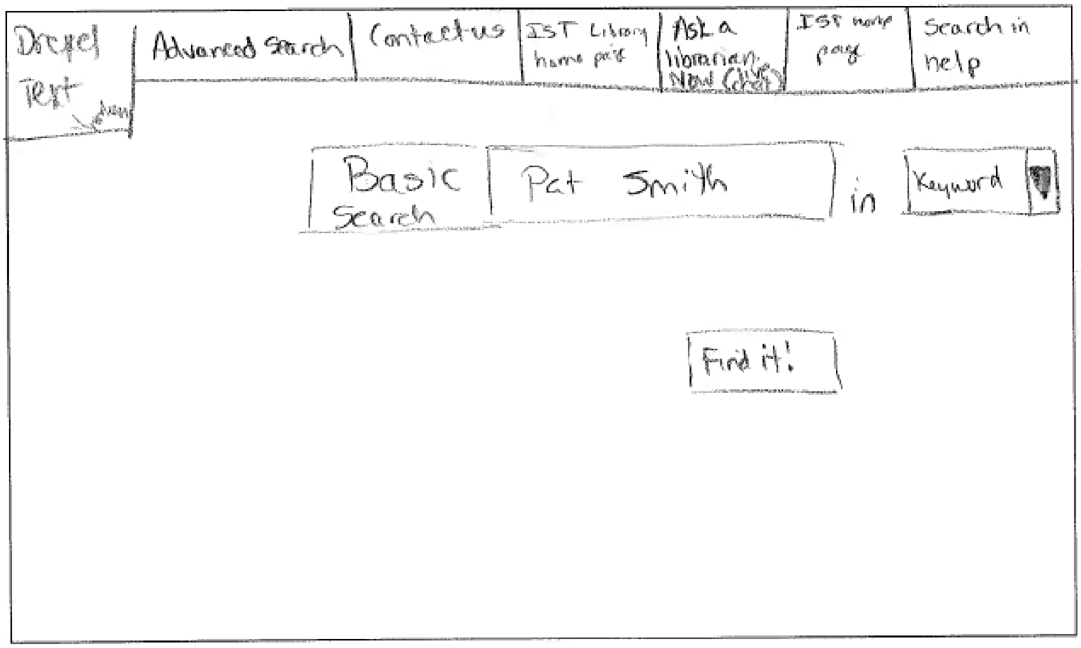
☐ Introduction to IST by PAT SMITH

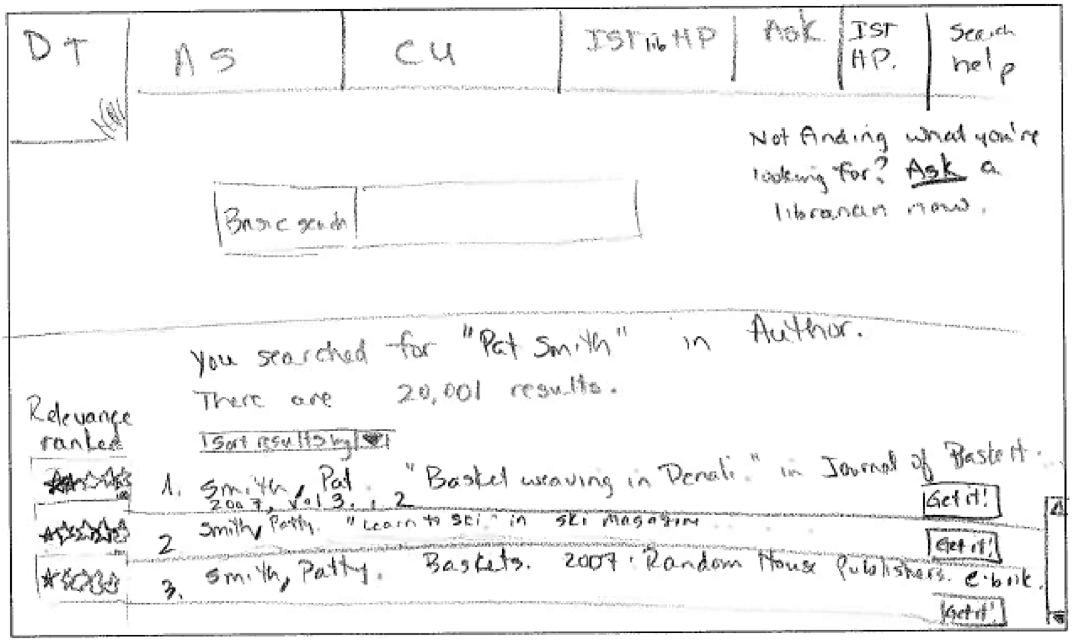
"Blurbz ahoy!"

☐ IST for Gerbil brained noobs by PAT SMITH

"Blurb"

NEXT PAGE ->





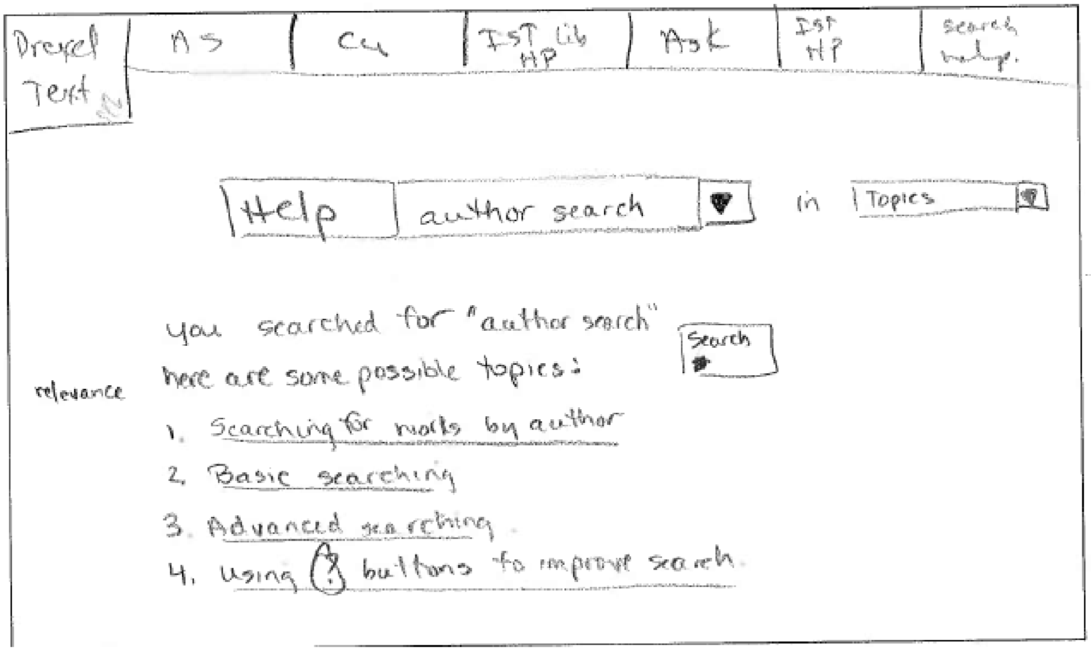
Not finding what you're looking for? Ask a librarian now.

You searched for "Pat Smith" in Author.
There are 20,001 results.

1. Smith, Pat. "Basket weaving in Denali." in Journal of Basket. 2007, Vol 3, 2

2 Smith, Patty. "Learn to ski." in ski Magazine

3. Smith, Patty. Baskets. 2007. Random House Publishers. e-book.



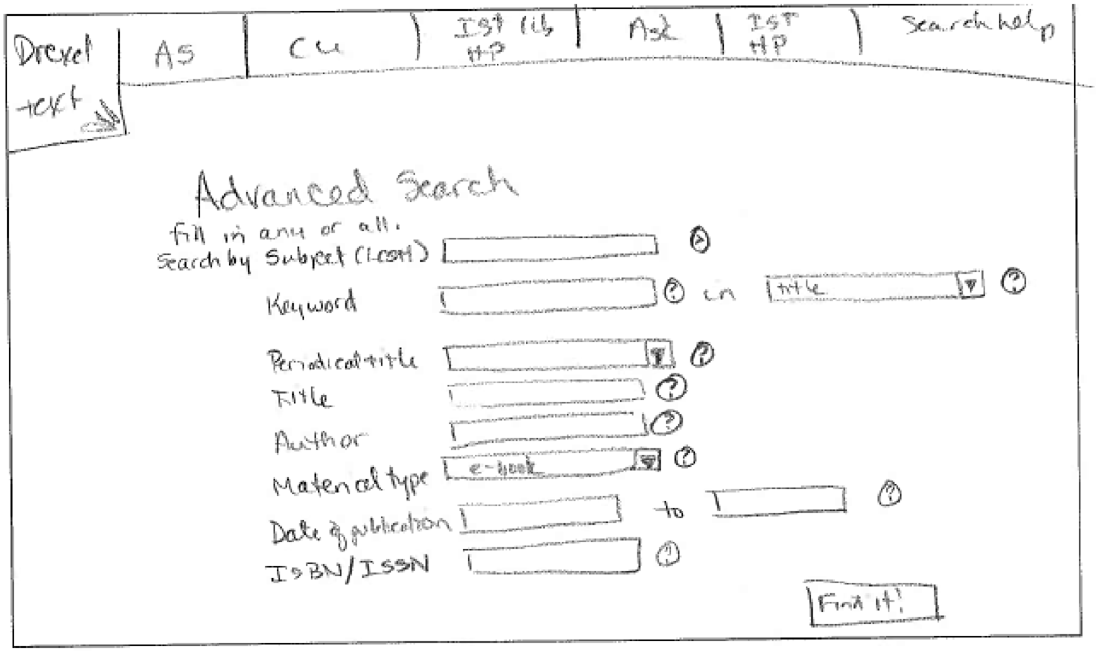
Help | author search ▼ in Topics ▼

relevance

You searched for "author search"

here are some possible topics:

1. Searching for works by author
2. Basic searching
3. Advanced searching
4. Using ? buttons to improve search.



Advanced Search

Fill in any or all.

Search by Subject (LCSH) ?

Keyword ? in title ▼ ?

Periodical title ▼ ?

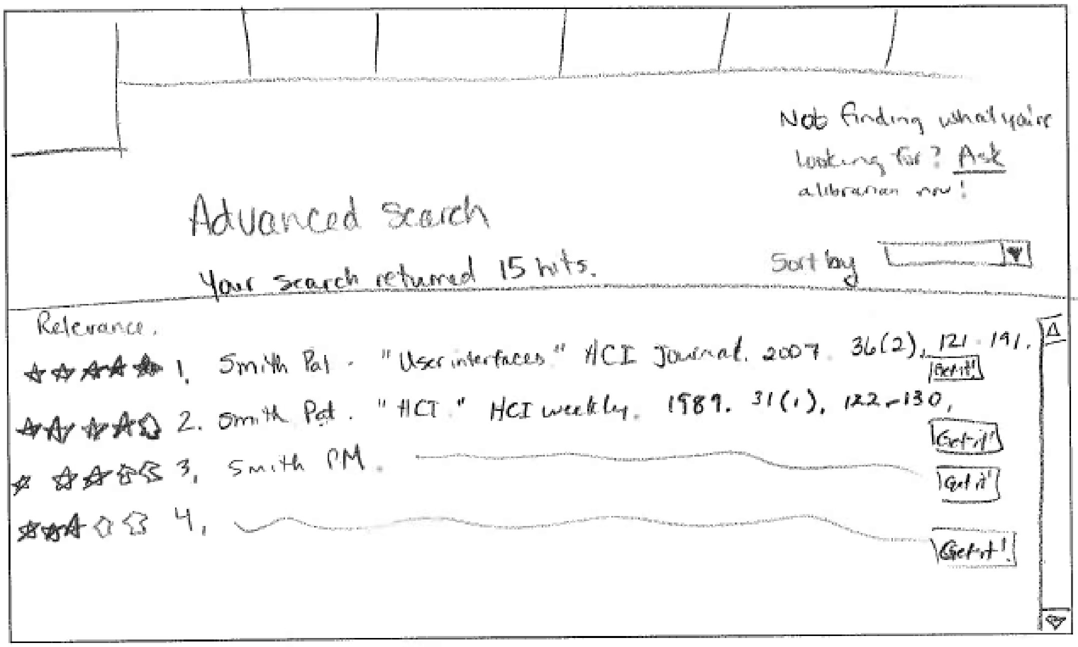
Title ?

Author ?

Material type e-book ▼ ?

Date of publication to ?

ISBN/ISSN ?



Not finding what you're looking for? Ask a librarian now!

Advanced Search

Your search returned 15 hits.

★★★★★ 1. Smith Pat. "User interfaces" HCI Journal. 2007. 36(2), 121-191.

★★★★☆ 2. Smith Pat. "HCI." HCI weekly. 1989. 31(1), 122-130,

★★★☆☆ 3. Smith PM.

★★★☆☆ 4.

Appendix N. Results of Intraclass Correlation using SPSS

Table N – 1. Item Statistics

	Mean	Std. Deviation	N
Rater1	3.89	1.45	208
Rater2	3.81	1.32	208

Table N – 2. Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.82	.82	2

Table N – 3. Intraclass Correlation Coefficient

	Intraclass Correlation (a)	95% Confidence Interval		F Test with True Value 0			
		Lower Bound	Upper Bound	Value	df1	df2	Sig
Single Measures	.69(b)	.62	.76	5.51	207.0	207	.00
Average Measures	.82(c)	.76	.86	5.51	207.0	207	.00

Appendix O. Assessing Normality using SPSS

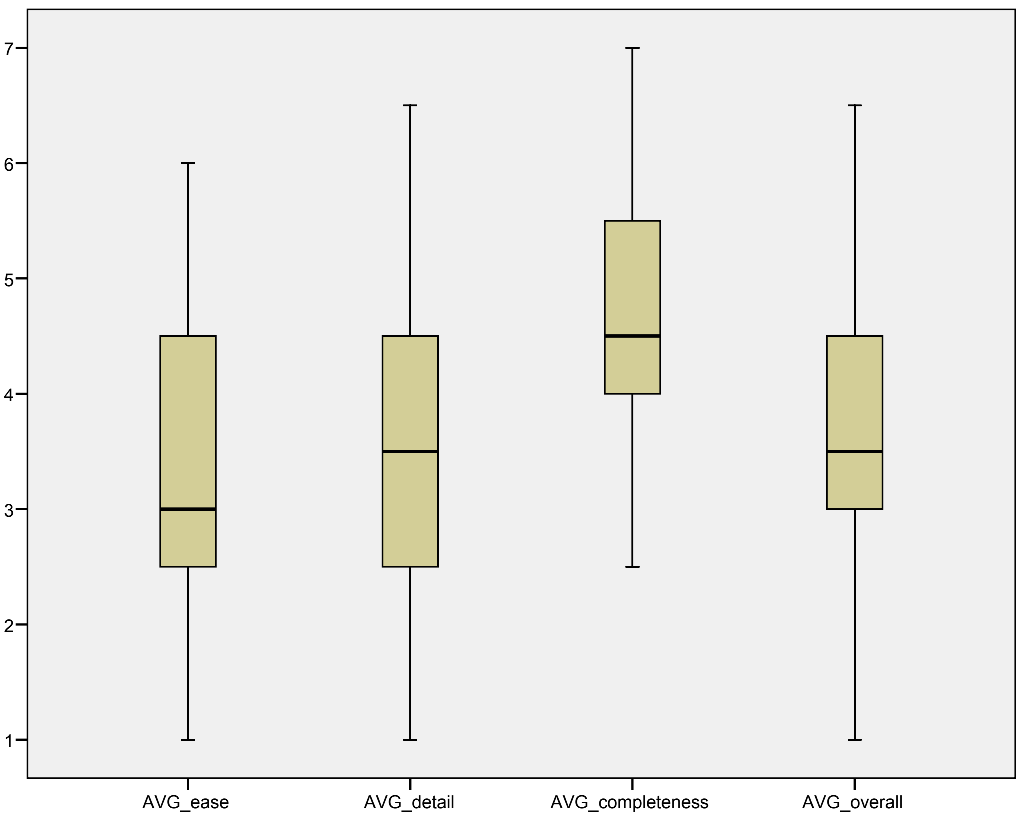


Figure O- 1. Box plots

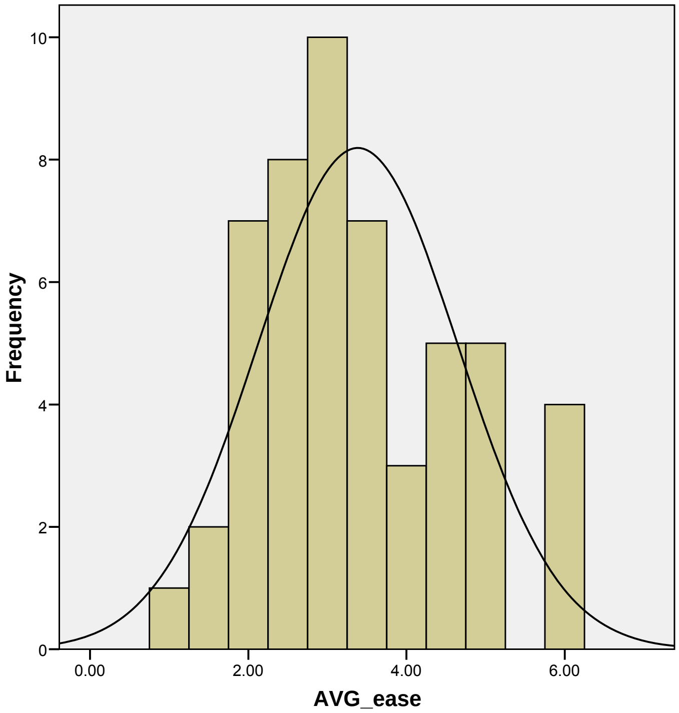


Figure O – 2. Ease of Use Histogram with Normal Distribution

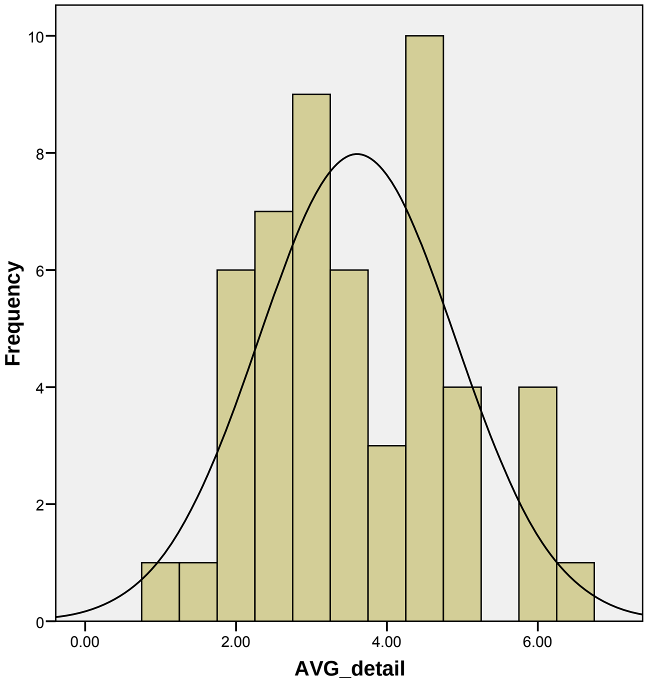


Figure O – 3. Average Detail Histogram with Normal Distribution

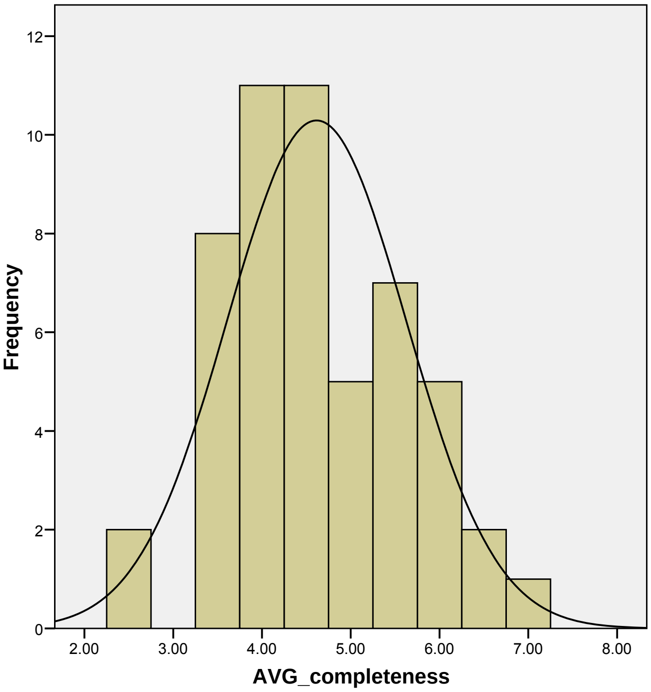


Figure O – 4. Average Completeness Histogram with Normal Distribution

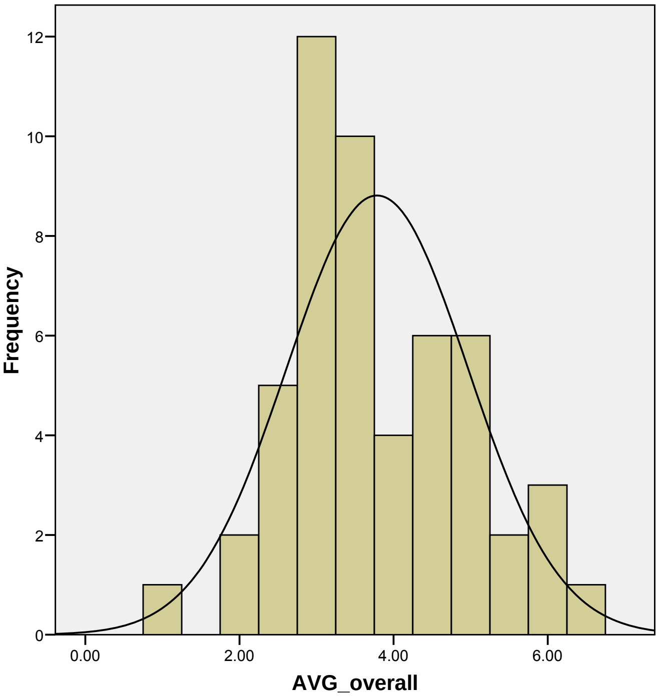


Figure O – 5. Average overall quality Histogram with Normal Distribution

Appendix P. Analyses for Research Questions using SPSS

Table P – 1. Correlations of Dependent Variables

		Ease of Use	Detail	Completeness	Overall Quality
Ease of Use	Pearson Correlation	1	.93(**)	.87(**)	.97(**)
	Sig. (2-tailed)		.00	.00	.00
	N	52	52	52	52
Detail	Pearson Correlation	.93(**)	1	.82(**)	.95(**)
	Sig. (2-tailed)	.00		.00	.00
	N	52	52	52	52
Completeness	Pearson Correlation	.87(**)	.82(**)	1	.90(**)
	Sig. (2-tailed)	.00	.00		.00
	N	52	52	52	52
Overall Quality	Pearson Correlation	.97(**)	.95(**)	.90(**)	1
	Sig. (2-tailed)	.00	.00	.00	
	N	52	52	52	52

** Correlation is significant at the 0.01 level (2-tailed).

ANOVA Ease of Use

Table P – 2. Descriptive Statistics Dependent Variable: Ease of Use

condition	Mean	Std. Deviation	N
Pattern	3.53	1.28	17
Guidelines	3.19	1.23	18
Control	3.44	1.34	17
Total	3.38	1.27	52

Table P – 3. Levene's Test of Equality of Error Variances Dependent Variable: Ease of Use

F	df1	df2	Sig.
.204	2	49	.82

Tests the null hypothesis that the error variance of the dependent variable is equal across groups.
a Design: Intercept+condition

Table P – 4. Tests of Between-Subjects Effects Dependent Variable: Ease of Use

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared	Noncent. Parameter	Observed Power(a)
Corrected Model	1.06(b)	2	.53	.32	.73	.01	.64	.10
Intercept	596.57	1	596.57	362.03	.000	.88	362.03	1.00
condition	1.06	2	.53	.32	.73	.01	.64	.10
Error	80.75	49	1.65					
Total	677.50	52						
Corrected Total	81.81	51						

a Computed using alpha = .05
b R Squared = .013 (Adjusted R Squared = -.027)

ANOVA Detail

Table P – 5. Descriptive Statistics Dependent Variable: Detail

condition	Mean	Std. Deviation	N
Pattern	3.82	1.30	17
Guidelines	3.47	1.10	18
Control	3.53	1.53	17
Total	3.60	1.30	52

Table P – 6. Levene's Test of Equality of Error Variances, Dependent Variable: Detail

F	df1	df2	Sig.
1.448	2	49	.25

Tests the null hypothesis that the error variance of the dependent variable is equal across groups.
a Design: Intercept+condition

Table P – 7. Tests of Between-Subjects Effects Dependent Variable: Detail

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared	Noncent. Parameter	Observed Power(a)
Corrected Model	1.23(b)	2	.61	.35	.70	.01	.71	.10
Intercept	676.57	1	676.57	390.29	.00	.89	390.29	1.00
condition	1.23	2	.61	.35	.70	.01	.71	.10
Error	84.94	49	1.73					
Total	762.25	52						
Corrected Total	86.17	51						

a Computed using alpha = .05
b R Squared = .014 (Adjusted R Squared = -.026)

ANOVA Completeness

Table P – 8. Descriptive Statistics Dependent Variable: Completeness

condition	Mean	Std. Deviation	N
Pattern	4.88	.98	17
Guidelines	4.31	.99	18
Control	4.68	1.03	17
Total	4.62	1.01	52

Table P -9. Levene's Test of Equality of Error Variances, Dependent Variable: Completeness

F	df1	df2	Sig.
.065	2	49	.94

Tests the null hypothesis that the error variance of the dependent variable is equal across groups.
a Design: Intercept+condition

Table P -10. Tests of Between-Subjects Effects Dependent Variable: Completeness

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared	Noncent. Parameter	Observed Power(a)
Corrected Model	3.00(b)	2	1.50	1.51	.232	.06	3.02	.31
Intercept	1109.80	1	1109.80	1114.24	.000	.97	1114.24	1.00
condition	3.00	2	1.50	1.51	.232	.06	3.02	.31
Error	48.81	49	.99					
Total	1159.50	52						
Corrected Total	51.81	51						

a Computed using alpha = .05
b R Squared = .058 (Adjusted R Squared = .020)

ANOVA Overall Quality

Table P – 11. Descriptive Statistics Dependent Variable: Overall Quality

condition	Mean	Std. Deviation	N
Pattern	4.00	1.13	17
Guidelines	3.56	1.17	18
Control	3.82	1.25	17
Total	3.79	1.18	52

Table P – 12. Levene's Test of Equality of Error Variances, Dependent Variable: Overall Quality

F	df1	df2	Sig.
.295	2	49	.75

Tests the null hypothesis that the error variance of the dependent variable is equal across groups.

Table P – 13. Tests of Between-Subjects Effects Dependent Variable: Overall Quality

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared	Noncent. Parameter	Observed Power(a)
Corrected Model	1.76(b)	2	.88	.63	.54	.03	1.25	.15
Intercept	747.58	1	747.58	531.55	.00	.92	531.55	1.00
condition	1.76	2	.88	.63	.54	.03	1.25	.15
Error	68.92	49	1.41					
Total	817.00	52						
Corrected Total	70.67	51						

a Computed using alpha = .05
b R Squared = .025 (Adjusted R Squared = -.015)

ANOVA Time

Table P – 14. Descriptive Statistics Dependent Variable: Design Time

condition	Mean	Std. Deviation	N
Pattern	21.00	7.46	17
Guidelines	23.39	13.55	18
Control	19.59	8.14	17
Total	21.37	10.10	52

Table P – 15. Tests of Between-Subjects Effects Dependent Variable: Design Time

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared	Noncent. Parameter	Observed Power
Corrected Model	129.66[a]	2	64.83	.63	.54	.03	1.25	.15
Intercept	23631.70	1	23631.70	228.38	.000	.82	228.38	1.00
condition	129.66	2	64.83	.63	.54	.03	1.25	.15
Error	5070.40	49	103.48					
Total	28937.00	52						
Corrected Total	5200.06	51						

a. R Squared= .025 (Adjusted R Squared = -.015)

b. Computed using alpha = .05

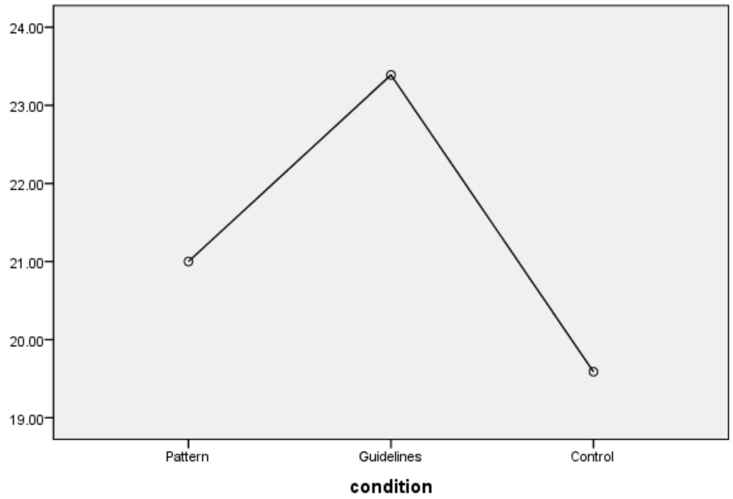


Figure P – 1. Means Plot Dependent Variable: Design Time

Table P – 16. Descriptive Statistics All Dependent Variables

	Mean	Std. Deviation	N
Design time	21.37	10.10	52
Ease of use	3.39	1.27	52
Detail	3.61	1.30	52
Completeness	4.62	1.01	52
Overall Quality	3.79	1.18	52

Table P – 17. Correlations between Time and Quality Measures

		Design time
Ease of use	Pearson Correlation	.40**
	Sig. (2-tailed)	.00
	N	52
Detail	Pearson Correlation	.36**
	Sig. (2-tailed)	.01
	N	52
Completeness	Pearson Correlation	.35*
	Sig. (2-tailed)	.01
	N	52
Overall Quality	Pearson Correlation	.41**
	Sig. (2-tailed)	.00
	N	52

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

Factor Analysis Orthogonal Rotation (varimax in SPSS)

Table P – 18. Total Variance Explained PCA (Orthogonal Rotation)

	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings		
Component	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	15.22	39.02	39.02	15.21	39.02	39.02	11.40	29.24	29.24
2	6.23	15.98	55.01	6.23	15.98	55.01	6.06	15.53	44.76
3	5.15	13.21	68.21	5.15	13.21	68.21	5.95	15.27	60.03
4	2.68	6.87	75.08	2.68	6.87	75.08	5.12	13.12	73.15
5	1.28	3.28	78.37	1.28	3.28	78.37	2.04	5.22	78.37
6	.68	1.75	80.11						
7	.61	1.56	81.67						
8	.55	1.41	83.08						
:	:	:	:	:	:	:	:	:	:
36	.01	.03	99.99						
37	.00	.01	99.99						
38	.00	.00	100.00						

Extraction Method: Principal Component Analysis.

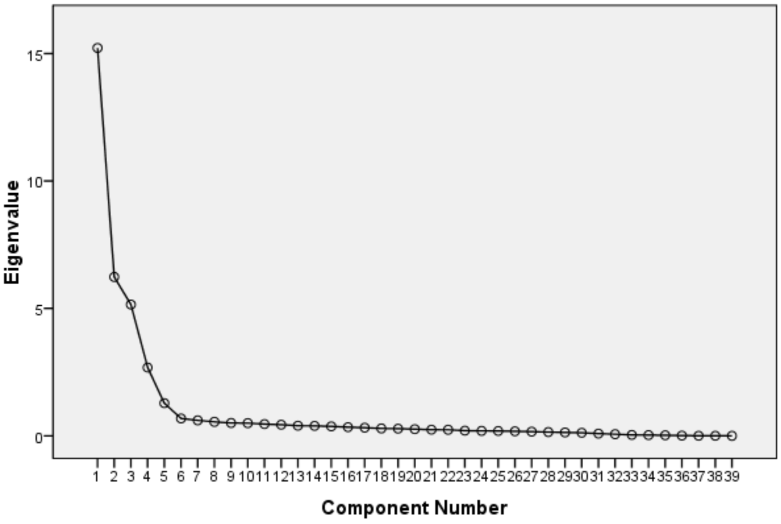


Figure P – 2. PCA Scree Plot (Orthogonal Rotation)

Table P – 19. PCA oblique factor rotation total variance explained

Compo	Initial Eigenvalues			Extraction Sums of Squared Loadings		
nent	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	15.219	39.022	39.022	15.219	39.022	39.022
2	6.233	15.983	55.005	6.233	15.983	55.005
3	5.150	13.206	68.211	5.150	13.206	68.211
4	2.679	6.869	75.080	2.679	6.869	75.080
5	1.281	3.284	78.365	1.281	3.284	78.365
6	.682	1.748	80.112			
7	.607	1.555	81.668			
8	.549	1.408	83.076			
:	:	:	:	:	:	:
37	.003	.009	99.998			
38	.001	.002	100.000			
39	-1.597E-16	-4.094E-16	100.000			
Extraction Method: Principal Component Analysis.						

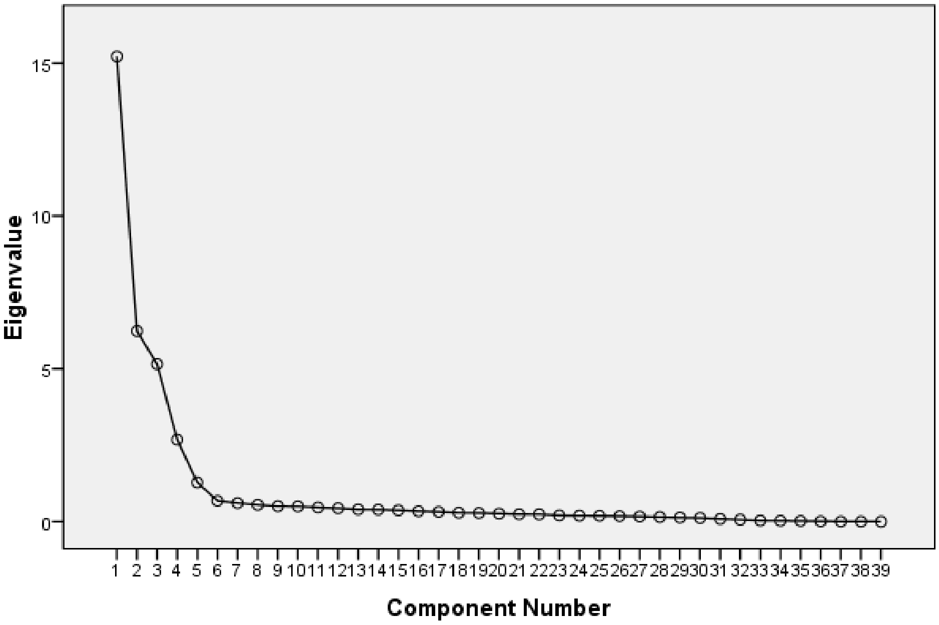


Figure P – 3. PCA Scree Plot (oblique factor rotation)

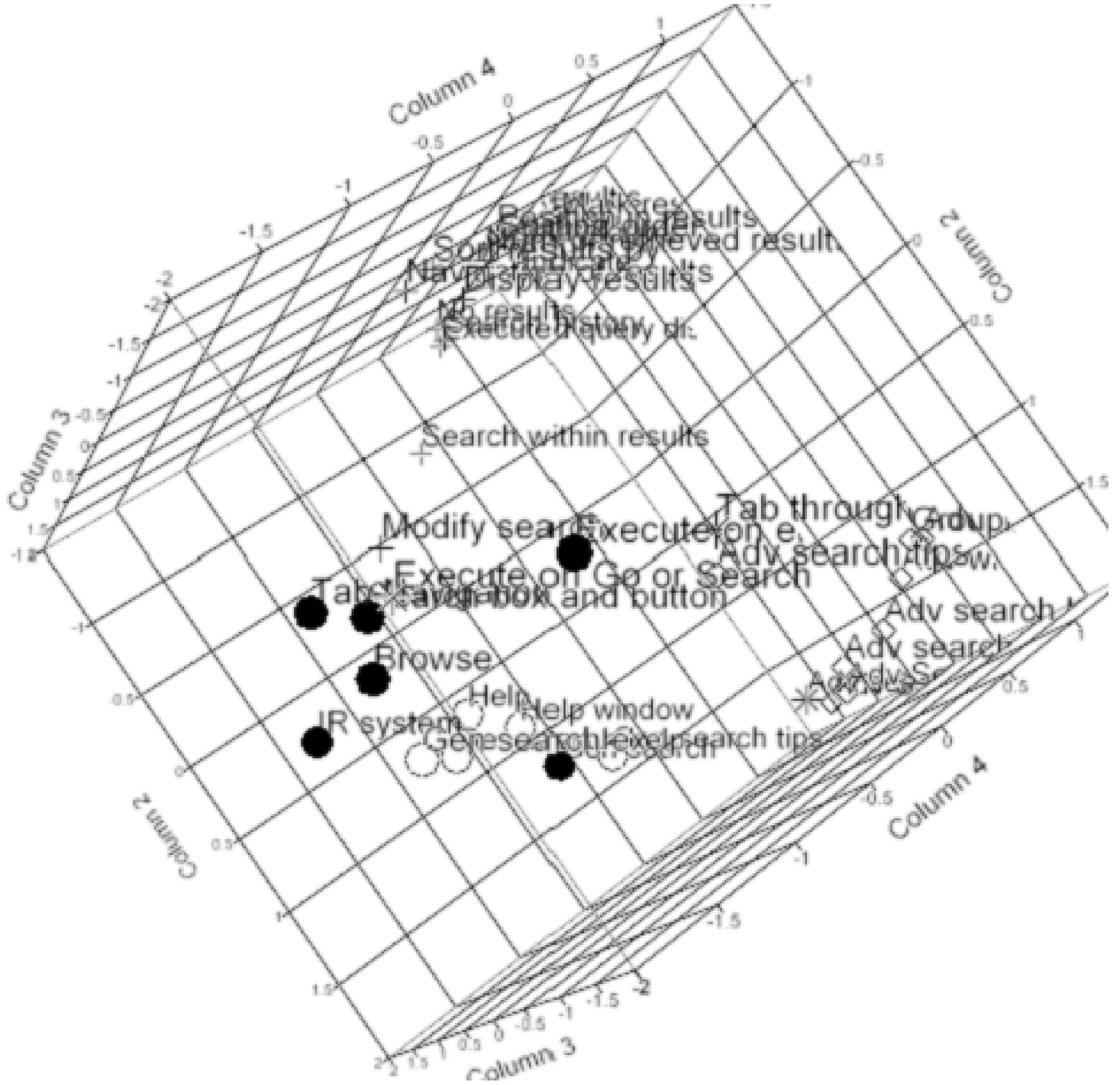


Figure P – 4. Three Dimensional MDS Map Rotation 1

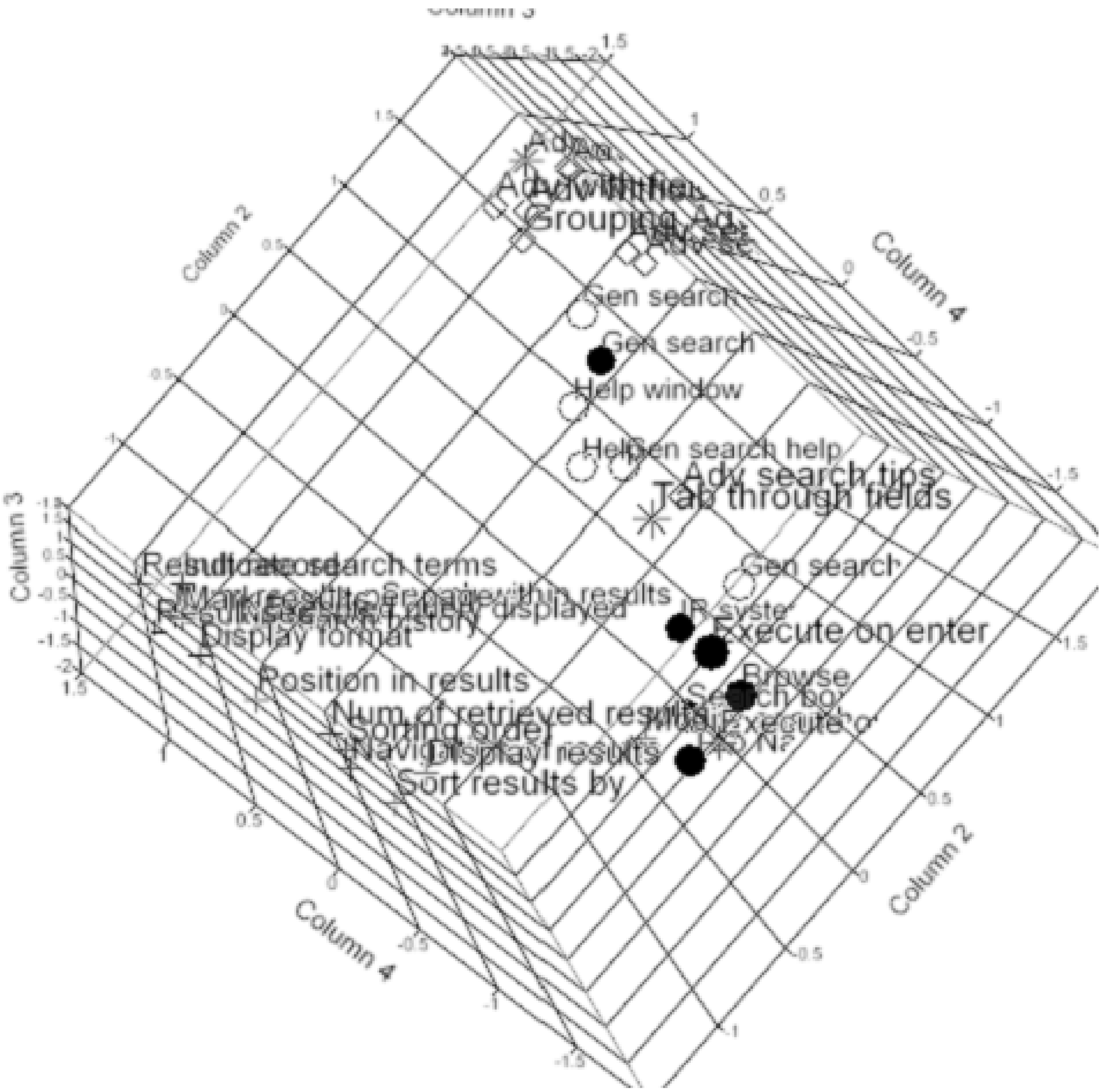


Figure P – 5. Three Dimensional MDS Map Rotation 2

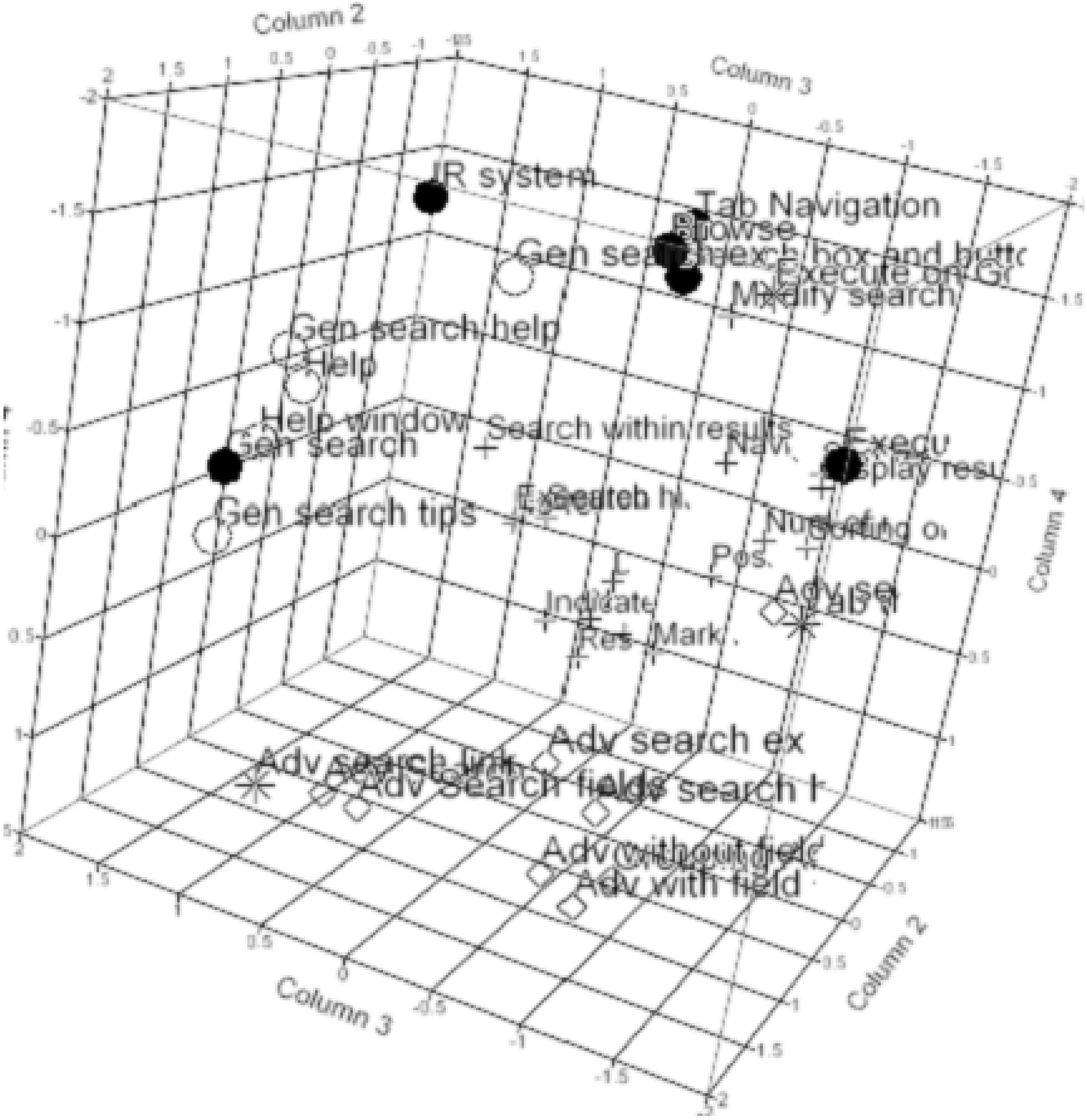


Figure P – 6. Three Dimensional MDS Map Rotation 3

Appendix Q. Patterns Identified in Interfaces

Table Q – 1. Patterns in IR systems and Search Engines and Participant Interfaces

No	Pattern Name	IR systems	Search Engines	Patterns Group	Guidelines Group	Control Group
		% systems used pattern	% systems used pattern	% participants used pattern	% participants used pattern	% participants used pattern
20	Help	100%	100%	100%	100%	100%
23	IR system	100%	100%	100%	100%	100%
15	General search	97%	100%	100%	100%	100%
33	Search box and button	97%	100%	100%	100%	100%
1	Advanced Search	97%	100%	100%	89%	100%
11	Display results	100%	100%	100%	89%	94%
8	Adv Search without field codes	80%	0%	88%	83%	76%
5	Advanced search link	20%	40%	82%	83%	76%
14	Executed search query displayed	30%	100%	65%	50%	71%
3	Advanced Search fields	93%	100%	94%	83%	65%
31	Result record	93%	100%	59%	28%	65%
17	General search help	87%	100%	35%	56%	59%
4	Advanced search help	83%	100%	35%	50%	47%
26	Navigation of results pages	87%	100%	65%	22%	47%
18	General search tips	77%	80%	71%	39%	35%
32	Result set	100%	100%	59%	28%	35%
29	Number of retrieved results	100%	100%	47%	28%	35%
30	Position in results	90%	100%	41%	17%	29%
21	Help window	67%	0%	47%	17%	18%
6	Advanced search tips	57%	80%	12%	0%	18%
36	Sort results by	90%	0%	53%	11%	12%
22	Indicate search terms in result set	30%	100%	29%	11%	12%
38	Tab Navigation	70%	0%	18%	11%	12%
13	Execute search on enter or go	97%	100%	0%	0%	12%
24	Mark results	77%	20%	41%	22%	6%
10	Display format	37%	0%	41%	6%	6%
9	Browse	60%	0%	24%	6%	6%
28	Number of results per page	60%	100%	12%	6%	6%

Table Q – 1. Patterns in IR systems and Search Engines and Participant Interfaces (Continued)

No	Pattern Name	IR systems	Search Engines	Patterns Group	Guidelines Group	Control Group
		% systems used pattern	% systems used pattern	% participants used pattern	% participants used pattern	% participants used pattern
16	General search example	70%	100%	6%	6%	6%
35	Search within results	47%	20%	29%	0%	6%
25	Modify search	67%	100%	29%	22%	0%
2	Advanced search example	60%	80%	6%	11%	0%
37	Sorting order	13%	0%	0%	6%	0%
7	Advanced search with field codes	33%	0%	12%	0%	0%
34	Search history	63%	20%	6%	0%	0%
39	Tab through fields	90%	100%	0%	0%	0%
19	Grouping Advanced Search fields	87%	100%	0%	0%	0%
27	No results	77%	100%	0%	0%	0%
12	Execute search on Go or Search	23%	20%	0%	0%	0%

ANOVA Patterns Used in Participant's Interfaces

Table Q – 2. Levene's Test of Equality of Error Variances Dependent Variable: Number of Patterns

F	df1	df2	Sig.
1.252	2	49	.30

Tests the null hypothesis that the error variance of the dependent variable is equal across groups.

a. Design: Intercept + condition

Table Q – 3. Tests of Between Subjects Effects Dependent Variable: Number of Patterns

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared	Noncent. Parameter	Observed Power[b]
Corrected Model	180.41[a]	2	90.20	7.10	.00	.23	14.21	.92
Intercept	10857.87	1	10857.87	854.97	.00	.95	854.97	1.00
condition	180.41	2	90.20	7.10	.00	.23	14.21	.92
Error	622.29	49	12.70					
Total	11620.00	52						
Corrected Total	802.69	51						

a. R Squared = .225 (Adjusted R Squared = .193)

b. Computed using alpha = .05

Table Q– 4. Post hoc analyses using Tukey (HSD) post hoc criterion

	(I) condition	(J) condition	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	Upper Bound
Tukey HSD	Pattern	Guidelines	4.28*	1.21	.00	1.37	7.19
		Control	3.53*	1.22	.02	.58	6.48
	Guidelines	Pattern	-4.28*	1.21	.00	-7.20	-1.37
		Control	-.75	1.21	.81	-3.66	2.16
	Control	Pattern	-3.53*	1.22	.02	-6.48	-.58
		Guidelines	.75	1.21	.81	-2.16	3.66

Based on observed means.

The error term is Mean Square(Error) = 12.700.

*. The mean difference is significant at the .05 level.

Table Q – 5. Correlations between number of patterns used in participant designs and quality ratings

		Number of patterns	Ease of Use	Level of Detail	Completeness	Overall Quality
Number of patterns	Pearson Correlation	1.00	.48**	.55**	.50**	.54**
	Sig. (2-tailed)		.00	.00	.00	.00
	N	52.00	51	51	51	51

**. Correlation is significant at the 0.01 level (2-tailed).

Appendix R. Evaluation Form for Ease of Use of IR Systems and Search Engines

IR system

URL

Instructions

Please make one pass through the system to familiarize yourself with the interface.

Please make a second pass through the interface and execute two queries:

1. A general search query for: patterns
2. An advanced search query for: patterns AND interaction

Please rate the overall ease of use of the system.

Ease of Use – How easy is it to use the system interfaces?

Ease of use	**Not at all** **Easy to use**						**Very easy** **to use**
	1	2	3	4	5	6	7

Comments:

VITA

Christine Elizabeth Wania

Education

Ph.D. in Information Studies College of Information Science and Technology, Drexel University	2008
B.S. in Computer Science State University of New York, College at Brockport	2003

Selected Publications

Atwood, M.E. and Wania, C.E. 2008. Design: Describing the Blind Men’s Elephant. Proceedings European Conference on Information Systems ECIS 2008, National University of Ireland, Galway.

Wania, C.E. and Atwood, M.E. 2007. aIRPLane: An Information Retrieval Pattern Language. CHI '07 extended abstracts on Human factors in computing systems, 2741-2746.

Robertson, S., Wania, C.E. and Park S.J. 2007. An Observational Study of Voters on the Internet. Proceedings of the 40th Annual Hawaii International Conference on System Sciences, 90.

Wania, C.E., Atwood, M.E. and McCain, K.W. 2006. How do Design and Evaluation Interrelate in Human Computer Interaction (HCI) Research? Proceedings of the 6th ACM conference on Designing Interactive Systems 2006: 90-98.

Silverstein, S. M., Wania, C.E. and Chen, Y. 2006. Access Patterns to a website on Healthcare IT Failure. American Medical Informatics Association Annual Symposium Proceedings, 1095.

Wania, C.E., Atwood, M.E. and McCain, K.W. 2006. Mapping the field of Human Computer Interaction. Proceedings ASIS&T 2006 Annual Meeting.

Scime, A. and Wania, C.E. 2005. Computing Curricula: A Comparison of Models. International Journal of Information and Communication Technology Education, 1(2) 1-18.

Selected Awards and Honors

Phoebe W. Haas Fellow, College of Information Science and Technology, Drexel University, 2007-2008

Best Paper Nominee: Robertson, S., Wania, C.E. and Park S.J. An Observational Study of Voters on the Internet. HICSS 2007: 90