Evaluating the Impact of a Pattern Structure on
Communicating Interaction Design Advice

A Thesis
Submitted to the Faculty
Of
Drexel University
by
Sunil George Abraham
in partial fulfillment of the
requirements for the degree
of
Doctor of Philosophy
Dec 2011
ACKNOWLEDGEMENTS

It has been a long and exhausting journey. For me, the PhD experience was deeply enriching, but what I hold dearest are the experiences, and people who have informed & guided me along the way.

All this would not be possible without the mentorship, and above all, patience of my dissertation chair, Dr. Michael Atwood. His subtle and not so subtle inquiries about my progress have kept me moving along. Most striking of his qualities is the ability to allow students to ask and pursue their own research questions, and never giving a direct answer. Only because finding my own path in this maze of research is part of the learning, and I am glad he adopted this approach. As he might still say, if you know the answers upfront to any research question, it’s not really research, is it?

I also want to thank my committee members, Dr. Susan Wiedenbeck, Dr. Denise Agosto, Dr. Rosina Weber, and Dr. Mark Gross. I admit that it was easy to read and critique my verbose writing, but thankfully all my committee members took the time and patience to give me meaningful and timely feedback, which was critical to finishing this dissertation. All this while juggling their own very hectic academic careers!

I do not want to forget the contribution of the iSchool at Drexel, as they made it possible for me to finish my doctoral study and research. The iSchool funded my education through research and teaching experiences, and I will always see their contribution as more than purely financial assistance. Being able to work closely with faculty members is essential to learning to be a PhD.

And finally, it’s always taxing for the student’s families who had to deal with a frantic, and often impatient, doctoral student. So a big thanks goes to my parents, sister, and my wife (Anu) for sticking it through with me.

If you run in to me on the street, and happen ask me whether the PhD was worth all the effort, I will always respond with a resounding yes! It’s not about the degree and title; it’s really about the learning.
# TABLE OF CONTENTS

LIST OF TABLES ........................................................................................................... IV

LIST OF FIGURES ......................................................................................................... VII

ABSTRACT ..................................................................................................................... IX

CHAPTER 1: INTRODUCTION ....................................................................................... 2

MOTIVATION & CONTRIBUTIONS ................................................................................. 3

CHAPTER 2: LITERATURE REVIEW ............................................................................. 6

2.1 INTRODUCTION ..................................................................................................... 6
2.2 DESIGN METHODOLOGY AND HCI .................................................................... 6
2.3 PATTERN CONSTRUCT IN ARCHITECTURE ....................................................... 7
2.4 PATTERN CONSTRUCT IN HCI ........................................................................... 10
2.5 REACTION TO PATTERNS .................................................................................... 14
2.5.1 Pattern Skeptics Within the Architecture Community .................................... 16
2.5.2 Interpreting & Adopting Pattern Concept By Other Design Communities ........ 18
2.6 PROMISE OF PATTERNS FOR HCI ..................................................................... 21
2.6.1 Patterns as Lingua Franca ............................................................................... 22
2.6.2 Patterns for Capturing and Communicating Design Experience ..................... 24
2.6.3 Patterns to Enable Knowledge Re-use ............................................................ 25
2.6.4 Patterns as Design Rationale ........................................................................... 27
2.7 PROBLEMS OR CHALLENGES FOR PATTERNS IN HCI ................................. 30
2.7.1 Common Format for Documenting Patterns in HCI ......................................... 32
2.7.2 Organizing Principle for Interaction Patterns: Pattern Languages or Collections 34
2.7.3 Tools to Aggregate, Search, and Use Patterns ............................................... 36
2.7.4 Evaluating Benefits of Interaction Patterns for Design of Interactive Systems ... 38
2.7.5 Summary Of Challenges and Agenda For Proposed Work .............................. 40
2.8 EMPIRICAL WORK ON PATTERNS IN HCI ......................................................... 40
2.8.1 Impact of Patterns in Education: Case Studies on Teaching With Patterns ...... 42
2.8.2 Impact of Patterns in Participatory Design: Case Studies Involving Users ........ 44
2.8.3 Impact of Patterns on Quality of the Designed Product: Controlled Studies .... 45
2.8.4 Summary Of Empirical Work ......................................................................... 49
2.9 SUMMARIZING EMPIRICAL WORK & RATIONALE FOR PROPOSED RESEARCH 51
2.9.1 Promises And Problems Revisited ................................................................. 52
2.9.2 Comments On Empirical Work ...................................................................... 56
2.9.2.1 Empirical Work on Teaching HCI Principles With Patterns ....................... 56
2.9.2.2 Empirical Work on Using Pattern in Participatory Design Setting ............. 57
2.9.2.3 Empirical Work Using Patterns and Impact on Quality of Product ............ 58
2.10 COMMUNICATING INTERACTION DESIGN ADVICE USING PATTERNS .......... 61
2.10.1 Pattern Structure in HCI .............................................................................. 63
2.10.2 Other Formats Used for Communicating Design Advice .............................. 66
2.10.3 Structuring Design Advice With Rationale .................................................. 68
2.10.4 Pattern-Like Structures Already in Use ....................................................... 70
2.10.5 Summary of Patterns and Communication ................................................... 72
2.10.5.1 Research Questions ............................................................................... 73
2.10.5.2 Comparing Pattern-Advice to Claims-Advice in the Study ....................... 75

CHAPTER 3: RESEARCH METHOD ............................................................................. 80

3.1 INTRODUCTION ................................................................................................... 80
## 3.2 Research Questions

- 3.2.1 Introduction
- 3.2.2 Opportunities to Discover Design Implications
- 3.2.3 Specifying and Scoping Context of Use
- 3.2.4 Assessing Maturity of Advice
- 3.2.5 Seedsing Re-Design

## 3.3 Participants

- 3.3.1 Introduction
- 3.3.2 Participant Population Overview & Data Collection
- 3.3.3 Summary of Results
- 3.3.4 RQ: Impact on the Effectiveness of Design Advice
- 3.3.5 Relationship between Dependent Variables
- 3.3.6 Summary of Findings for Worthiness of Design Advice

## 3.4 Research Design

- 3.4.1 Independent Variable
- 3.4.2 Dependent Variables
- 3.4.3 Materials & Instruments
- 3.4.4 Procedure for Design Activity
- 3.4.5 Procedure for Expert Evaluation

## 4. Results

- 4.1 Introduction
- 4.2 Participant Population Overview & Data Collection
- 4.3 Summary of Results
- 4.4 RQ: Impact on the Effectiveness of Design Advice
- 4.5 RQ II: Impact of Advice Structure on Design Time
- 4.6 Additional Findings from Content-Analysis

## 5. Discussion of Findings

- 5.1 Summary of Findings
- 5.2 Communicating Design Advice
- 5.2.1 Cons in Design Advice
- 5.2.2 Trade-Offs in Design as a Shared Vocabulary

### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2 Research Questions</td>
<td>81</td>
</tr>
<tr>
<td>3.3 Participants</td>
<td>82</td>
</tr>
<tr>
<td>3.4 Research Design</td>
<td>82</td>
</tr>
<tr>
<td>3.4.1 Independent Variable</td>
<td>83</td>
</tr>
<tr>
<td>3.4.2 Dependent Variables</td>
<td>83</td>
</tr>
<tr>
<td>3.4.3 Materials &amp; Instruments</td>
<td>86</td>
</tr>
<tr>
<td>3.4.3.1 Pre-task Questionnaire</td>
<td>87</td>
</tr>
<tr>
<td>3.4.3.2 Practice Task</td>
<td>88</td>
</tr>
<tr>
<td>3.4.3.3 Design Task</td>
<td>88</td>
</tr>
<tr>
<td>3.4.3.4 Structure of Design Advice</td>
<td>92</td>
</tr>
<tr>
<td>3.4.3.5 Post-task Questionnaire</td>
<td>94</td>
</tr>
<tr>
<td>3.4.3.6 Evaluation Worksheet</td>
<td>95</td>
</tr>
<tr>
<td>3.4.4 Procedure for Design Activity</td>
<td>96</td>
</tr>
<tr>
<td>3.4.5 Procedure for Expert Evaluation</td>
<td>98</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>101</td>
</tr>
<tr>
<td>4.2 Participant Population Overview &amp; Data Collection</td>
<td>102</td>
</tr>
<tr>
<td>4.3 Summary of Results</td>
<td>107</td>
</tr>
<tr>
<td>4.4 RQ: Impact on the Effectiveness of Design Advice</td>
<td>108</td>
</tr>
<tr>
<td>4.4.1 Assessing Inter-rater Reliability</td>
<td>110</td>
</tr>
<tr>
<td>4.4.2 Exploring Data: Normality and Equal Variances</td>
<td>111</td>
</tr>
<tr>
<td>4.4.3 Relationship between Dependent Variables</td>
<td>113</td>
</tr>
<tr>
<td>4.4.4 Summary of Findings for Worthiness of Design Advice</td>
<td>121</td>
</tr>
<tr>
<td>4.5 RQ II: Impact of Advice Structure on Design Time</td>
<td>122</td>
</tr>
<tr>
<td>4.6 Additional Findings from Content-Analysis</td>
<td>125</td>
</tr>
<tr>
<td>4.6.1 Summary of Findings from Content Analysis</td>
<td>127</td>
</tr>
<tr>
<td>4.6.2 Impact of Advice Structure on Number of Words in Advice</td>
<td>128</td>
</tr>
<tr>
<td>4.6.3 Impact of Advice Structure on Context of Use Described</td>
<td>131</td>
</tr>
<tr>
<td>4.6.4 Impact of Advice Structure on Evaluative Comments Made</td>
<td>135</td>
</tr>
<tr>
<td>4.6.5 Impact of Advice Structure on Design Issues or Implications Considered</td>
<td>140</td>
</tr>
<tr>
<td>4.6.6 Summary of Content Analysis</td>
<td>143</td>
</tr>
<tr>
<td>5.1 Summary of Findings</td>
<td>147</td>
</tr>
<tr>
<td>5.1.1 Usefulness of Advice</td>
<td>148</td>
</tr>
<tr>
<td>5.1.2 Rationale Presented in Design Advice</td>
<td>150</td>
</tr>
<tr>
<td>5.1.3 Context of Use in Advice</td>
<td>150</td>
</tr>
<tr>
<td>5.2 Communicating Design Advice</td>
<td>151</td>
</tr>
<tr>
<td>5.2.1 Cons in Design Advice</td>
<td>152</td>
</tr>
<tr>
<td>5.2.2 Trade-Offs in Design as a Shared Vocabulary</td>
<td>161</td>
</tr>
<tr>
<td>5.2.2.1 Interpreting Design Principle(s)</td>
<td>161</td>
</tr>
<tr>
<td>5.2.2.2 Applying Design Principle(s) in Practice</td>
<td>164</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 2-1 Comparing some common pattern formats in HCI .......................................................... 32
Table 2-2 Comparing problem statements presented by pattern authors in HCI compared to Alexander’s (1977) version ........................................................................................................... 33
Table 2-3 Showing the basic pattern structure of problem, solution, and context present in different versions of patterns in HCI ................................................................................................. 64
Table 2-4 Highlighting the similarities/differences between patterns, claims, QOC and PHI techniques with respect to intent and strengths of each approach ..................................................................................... 78
Table 2-5 Highlighting the similarities/differences between patterns, claims, QOC and PHI structure .................................................................................................................................................... 79
Table 3-1 Table showing the three levels of the independent variable, structure of design advice, and related treatment .................................................................................................................... 83
Table 3-2 Descriptive information provided alongside the menu options ......................................................................................................................... 92
Table 4-1 Experimental Design showing the distribution of participants and treatment ............ 103
Table 4-2 Distribution of participants’ educational background and age .................................... 103
Table 4-3 Distribution of participants’ experience working in interaction design and number of courses completed in HCI .................................................................................................................. 104
Table 4-4 Additional instructions provided to the participants based on the assigned condition 105
Table 4-5 Average time (in minutes) taken to complete the practice and design task for each experiment condition ........................................................................................................................................ 107
Table 4-6 Summary of results from statistical tests ........................................................................ 107
Table 4-7 Intra-class correlation for individual metrics and for the entire sample ............................ 111
Table 4-8 Mean, standard deviation and median for the dependent variables (ratings) .............. 111
Table 4-9 Checking for normality for context of use, rationale expressed, usefulness and overall quality before conducting statistical tests (ANOVA) ............................................................................. 112
Table 4-10 Correlation coefficients, significance, and strength of correlation between any two dependent variables ...................................................................................................................... 113
Table 4-11 Descriptive Statistics for Context of Use Described ...................................................... 116
Table 4-12 Descriptive Statistics for Rationale Expressed ............................................................... 117
Table 4-13 Descriptive Statistics for Usefulness ............................................................................. 119
Table 4-14 Descriptive Statistics for Overall Quality ...................................................................... 120
Table 4-15 Assessing normality for design time ............................................................................. 123
Table 4-16 Time taken to complete the experiment ....................................................................... 123
Table 4-17 Assessing correlation between design time and dependent variables in research question 1 .............................................................................................................................................. 125
Table 4-18 Summary of findings from Content Analysis. Significance at p<.05 level ............... 128
Table 4-19 Average number of words used to describe design decision and its standard deviation .............................................................................................................................................. 128
Table 4-20 Means and Standard Deviation for #Words used in the three conditions. .............. 130
Table 4-21 Correlations between the measures for worthiness of advice and number of words present in the design advice .............................................................................................................. 131
Table 4-22 Mean and Standard deviation for Context_Count and checking for Normality ........ 133
Table 4-23 Mean and Standard Deviation for Context_Count ........................................ 134
Table 4-24 Correlation between Context_Count and Context-of-Use Rating ................ 135
Table 4-25 Descriptive Statistics for Number of Evaluative Comments and tests for assessing normality. ........................................................................................................... 137
Table 4-26 Mean and Standard Deviation for Number of Evaluative Comments made by Condition ................................................................................................................ 138
Table 4-27 Correlation Coefficients and Variance Explained between number of evaluative comments and measures for goodness of design advice ........................................ 139
Table 4-28 Mean and Standard Deviation for number of issues identified; Testing for Normality using Skewness/Kurtosis and 5% trimmed means .................................................. 140
Table 4-29 Descriptive Statistics for Number of Design Issues Discussed .................. 142
Table 4-30 Correlation between # of design issues discussed and usefulness .............. 143
Table 4-31 Summarizing the relationships between the dependent measures in RQ1 and content analysis ............................................................................................................. 144
Table 5-1 Results from ANOVA for each dependent variable ..................................... 147
Table 5-2 Descriptive statistics showing means (M) and std. deviations (SD). A 7-pt Lickert type scale was used. Values in BOLD imply that is the highest value in that row ............. 148
Table 5-3 Summarizing the relationships between the dependent measures in RQ1 and Content Analysis ............................................................................................................. 149
Table 5-4 Total number of words present in design advice from three experiment conditions . 151
Table 5-5 Correlation between # of design issues discussed and usefulness ............... 152

Table 2-1 Comparing some common pattern formats in HCI ........................................ 32
Table 2-2 Comparing problem statements presented by pattern authors in HCI compared to Alexander’s (1977) version ................................................................. 33
Table 2-3 Showing the basic pattern structure of problem, solution, and context present in different versions of patterns in HCI ......................................................... 64
Table 2-4 Highlighting the similarities/differences between patterns, claims, QOC and PHI techniques with respect to intent and strengths of each approach ................................ 78
Table 2-5 Highlighting the similarities/differences between patterns, claims, QOC and PHI structure ............................................................................................................. 79
Table 3-1 Table showing the three levels of the independent variable, structure of design advice, and related treatment ................................................................. 83
Table 3-2 Descriptive information provided alongside the menu options ..................... 92
Table 4-1 Experimental Design showing the distribution of participants and treatment ..... 103
Table 4-2 Distribution of participants’ educational background and age ..................... 103
Table 4-3 Distribution of participants’ experience working in interaction design and number of courses completed in HCI ................................................................. 104
Table 4-4 Additional instructions provided to the participants based on the assigned condition 105
Table 4-5 Average time (in minutes) taken to complete the practice and design task for each experiment condition ......................................................... 107
Table 4-6 Summary of results from statistical tests .................................................. 107
Table 4-7 Intra-class correlation for individual metrics and for the entire sample ........... 111
Table 4-8 Mean, standard deviation and median for the dependent variables (ratings) .... 111
Table 4-9 Checking for normality for context of use, rationale expressed, usefulness and overall quality before conducting statistical tests (ANOVA) .......................................................... 112
Table 4-10 Correlation coefficients, significance, and strength of correlation between any two dependent variables. ........................................................................................................ 113
Table 4-11 Descriptive Statistics for Context of Use Described .......................................................... 116
Table 4-12 Descriptive Statistics for Rationale Expressed .......................................................... 117
Table 4-13 Descriptive Statistics for Usefulness .................................................................................. 119
Table 4-14 Descriptive Statistics for Overall Quality ........................................................................... 120
Table 4-15 Assessing normality for design time .............................................................................. 123
Table 4-16 Time taken to complete the experiment ........................................................................... 123
Table 4-17 Assessing correlation between design time and dependent variables in question 1 ....................................................................................................................................... 125
Table 4-18 Summary of findings from Content Analysis. Significance at p<.05 level ................. 128
Table 4-19 Average number of words used to describe design decision and its standard deviation ........................................................................................................................................ 128
Table 4-20 Means and Standard Deviation for #Words used in the three conditions. .......... 130
Table 4-21 Correlations between the measures for worthiness of advice and number of words present in the design advice. .............................................................................................. 131
Table 4-22 Mean and standard deviation for Context_Count and checking for Normality........ 133
Table 4-23 Mean and Standard Deviation for Context_Count ............................................................... 134
Table 4-24 Correlation between Context_Count and Context-of-Use Rating .................................. 135
Table 4-25 Descriptive Statistics for Number of Evaluative Comments and tests for assessing normality. ........................................................................................................................................ 137
Table 4-26 Mean and Standard Deviation for Number of Evaluative Comments made by Condition ........................................................................................................................................ 138
Table 4-27 Correlation Coefficients and Variance Explained between number of evaluative comments and measures for goodness of design advice .......................................................... 139
Table 4-28 Mean and Standard Deviation for number of issues identified; Testing for Normality using Skewness/Kurtosis and 5% trimmed means. ................................................................. 140
Table 4-29 Descriptive Statistics for Number of Design Issues Discussed ........................................ 142
Table 4-30 Correlation between # of design issues discussed and usefulness .................................. 143
Table 4-31 Summarizing the relationships between the dependent measures in RQ1 and content analysis ........................................................................................................................................ 144
Table 5-1 Results from ANOVA for each dependent variable ................................................................. 147
Table 5-2 Descriptive statistics showing means (M) and std. deviations (SD). A 7-pt Lickert type scale was used. Values in BOLD imply that is the highest value in that row. ................................. 148
Table 5-3 Summarizing the relationships between the dependent measures in RQ1 and Content Analysis ........................................................................................................................................ 149
Table 5-4 Total number of words present in design advice from three experiment conditions .. 151
Table 5-5 Correlation between # of design issues discussed and usefulness ................................... 152
LIST OF FIGURES

Figure 2–1. Graph showing the trend for citations received by Alexander et al. (1977) from (1972-2007) computed from science citation index (Web of Knowledge)................................................................. 20
Figure 2–2. Overview of motivations for patterns in HCI................................................................. 21
Figure 2–3. Example of a claim describing the advantages and disadvantages of a “pop-up style alert” adapted from (A. Fabian et al., 2006)........................................................................................................ 29
Figure 2–4. Overview of the challenges faced before realizing the promises made about patterns in HCI................................................................................................................................................. 31
Figure 2–5 Review of empirical work investigating the benefits of using patterns in design...... 41
Figure 2–6 Overview of empirical work conducted on patterns in HCI........................................ 42
Figure 2–7 Overview of research on patterns in HCI: promises, challenges and contributions made through empirical work, and refocusing on communication.................................................. 53
Figure 2–8 MDS map showing results of an author co-citation analysis, or the number of times two authors were cited together in one document, in the field of HCI (reproduced with permission of copyright owners) (1990-2004)................................................................. 66
Figure 2–9 An example log-in screen, which reminds the visitor the kind of information expected. ........................................................................................................................................... 67
Figure 3–1 Overview of the methodology used in this study. ......................................................... 87
Figure 3–2 Practice task asking participants to recommend an icon for the mobile phone task context............................................................................................................................................. 88
Figure 3–3 Part of the instructions provided to participants about the context of the design task. 89
Figure 3–4 A barebones interface showing the task area of a room planner ................................ 90
Figure 3–5 Three menu options are available to the participants: (a) Directory Tree Menu, (b) Drop-Down Menu, (c) Accordion Menu........................................................................................................... 91
Figure 3–6 Materials provided to participants as part of the design task for recommending a menu design for room planner interface. .................................................................................. 97
Figure 3–7 Materials provided to expert evaluators consisting of the reference sheet, 52 individual pieces of advice, and one good example of advice......................................................... 99
Figure 4–1 Overview of process followed to assess the inter-rater reliability between expert evaluators. An average of the rating from both evaluators was used in the statistical tests. ................................................. 109
Figure 4–2 Illustrating an example rating set where the judges were consistent, but not in agreement. ............................................................................................................................................ 110
Figure 4–3 Means plot for Context of Use Described........................................................................ 116
Figure 4–4 Means plot for Rationale............................................................................................... 117
Figure 4–5 Means plot Usefulness................................................................................................. 119
Figure 4–6 Means plot for Overall Quality.................................................................................... 120
Figure 4–7 Means plot for Design Time........................................................................................ 124
Figure 4–8 Means plot for number of words (#words) present in design advice .................... 130
Figure 4–9 Means plot for Context_Count (number of words coded describing context) ....... 134
Figure 4–10 Means plot for Number of Evaluative Comments made....................................... 138
Figure 4–11 Means plot for Number of Unique Design Issues Discussed............................... 142
Figure 5–1 Front Entrance to a home ................................................................. 156
Figure 5–2 Bedroom in a home ................................................................. 156
Figure 5–3 A room planner interface showing tabbed sections and drop down lists to organize and access products ................................................................. 157
Figure 5–4 A conceptual rendering of the pros and cons in the solution; in one, the forces are perfectly balanced (ideal solution), and in the other the forces are prioritized (visibility over complexity) ................................................................. 159
Figure 5–5 Conceptualizing the contribution of cons in an iterative evaluation-driven design lifecycle ................................................................. 160
Figure 5–6 Example A shows a traditional MS Word 2007 with the Ribbon interface. Example B shows a simple text editor in OS X, similar to the Wordpad application in the Windows OS ................................................................. 162
Figure 5–7 Example A is shows a common instance of MS Word; Example B shows the MS Word application with all the toolbars toggled visible ................................................................. 163
Figure 5–8 Illustrating the relationship between complexity and visibility principles in design practice. Value of 0 indicates a design were this principle has not been considered, and a value of 1 indicates when a principles has been perfectly realized ................................................................. 165
Figure 5–9 A graphical representation of Norman’s Gulfs of Execution and Evaluation and the various stages that help bridge these gulfs ................................................................. 169
Figure 5–10 Using design instances or examples of clear and “good” navigation being used on different websites from http:// patterntap.com/tap/collection/navigation ................................................................. 169
Figure 5–11 Each node represents a design principle, and each block represents a group of principles that have relationships defined that have been extracted from the solution instances. The size of the nodes indicates the priority assigned during trade-off ................................................................. 171
Figure 5–12 Design advice occupying a mid-level between principles and concrete design instances; the levels from bottom to top correspond to an increasing level of specificity of context. The icons on the right correspond to shareable (vocabulary), tangible (visual) and reusable advice ................................................................. 173
Figure 5–13 Illustrating the home layout design problem ................................................................. 175
Figure 5–14 A candidate solution for the given home-design problem ................................................................. 175
Figure 5–15 Comparing pattern-structure and claim-structure with aspects desired of design advice ................................................................. 179
Figure 5–16 Illustrating the enhanced structure, and how the independent pattern and claim structural elements participate in this proposed structure ................................................................. 181
Figure 5–17 Menus shown as part of 1) diagramming application and 2) Microsoft Word 2007 toolbar ................................................................. 185
Figure 5–18 Autofill shown as part of 1) an email client and 2) Firefox browser address bar ... 189
Figure 6–1 Comparing design guidance that include rationale with those which offer a set of “good” design instances; Breadcrumb Pattern (design instances) from http:// patterntap.com/tap/collection/breadcrumbs; Breadcrumb Pattern (design instances + rationale) from http:// www.welie.com/patterns/showPattern.php?patternID=crumbs ........ 199
ABSTRACT

This study reports findings from a controlled experiment evaluating the benefits of structuring design advice as patterns. Over the years, the pattern concept from architecture has become a native within the HCI community and its related discussions on sharing design knowledge. It is argued that the context-rich, and tangible, nature of patterns has contributed to its acceptance by the community. And despite a lack of empirical work demonstrating benefits of patterns to design or communication, pattern collections continue to grow in popularity and acceptance. This research responds to the call for much needed empirical work in the area of patterns in HCI, and explores the value of a pattern structure. Our findings suggest that it may be beneficial to combine the strengths (or differences) of each advice structure (pattern and claim) to yield a more robust structure to capture design advice. We arrive on these findings based on evaluating design advice produced using either a pattern or claim structure. Two expert judges using the following four criteria rated the prepared advice: context-of-use described, rationale provide, usefulness of advice, and overall quality.

We argue that while patterns may be "natural" approach exploited by design professionals, the discussion on what patterns may have to offer may has been corrupted by a narrow focus on patterns-as-result. We suggest that patterns need a renewed focus and rethink, but this time exploring the benefits of patterns-as-a-process. We also argue that trade-offs in context are integral to a design activity, and why design advice needs to be about resolving trade-offs, and not about specific usability goals. Furthermore, discussing cons as part of the advice can lead to identifying more number of participating design issues, help in scoping the context-of-use, reflect the maturity and confidence in given advice, and potentially seed re-design activity. Design advice, as proposed (i.e., unifying patterns and claims), has the potential to fulfill a gap between theory based design methods and concrete design instances.
CHAPTER 1: INTRODUCTION

The word design evokes many interpretations. It may not be surprising to hear one complement a product as a good design where design is a product attribute. Design has another interpretation. Atwood, McCain, and William’s (2002) sampling of definitions for design seem to converge on design is a process, not necessarily the product. The present research is concerned with this interpretation of design, and particularly, design of interactive systems.

A challenge for any design discipline is to make design experience explicit (Pemberton, 2000), to be shared with others within the community of practice, and often across disciplinary boundaries. This allows design methods (or research) to evolve, by building on the success and failures of others. Over the years, human-computer interaction (HCI) community has strived to make explicit what they know about designing interactive systems in different ways. Principles derived from studies of human cognition for understanding the capacities and limitations of end-users (Card et al., 1984; Norman, 1986); basic tenets for a user-centered design process (Gould & Lewis, 1985); HCI design lifecycle models (Mayhew, 1999); and specific methods for evaluating usability of interactive systems (Nielsen, 1994; Polson & Lewis, 1990) are some of key practices taught today in HCI.

Within the universe of HCI design and evaluation practices, patterns are a popular topic. The past few decades has witnessed growing interest in patterns in HCI. This is evident in the proliferation in the number of interaction patterns available online (e.g., Yahoo! Pattern library) and in print (Tidwell, 2005). An interaction pattern aims to capture the essence of a solution to a recurring interaction design problem in a specific context. Patterns in HCI draw their inspiration from patterns in architecture and urban design, introduced by Alexander et al. (1977). Software engineering community was the first to repurpose the pattern concept to capture best practices in object-oriented design, with the explicit goal of reusing these best practices (Gamma et al., 1995). The HCI community, however, expects more from patterns than just reuse (Dearden and Finlay,
2006). It has been argued that patterns in HCI have the potential to contribute to a common language for interaction design; a language that can be used by users and designers to communicate. The pattern structure for capturing HCI design experience has been asserted as an improvement over other forms of design guidance (e.g., guidelines & style guides).

Much has been promised of patterns in HCI: as lingua francas for design, enabling knowledge reuse, as a way to express design rationale, and as a structure to capture and communicate HCI design experience. The past few decades have revealed little empirical evidence demonstrating the value of patterns and pattern languages to the design process or designed product. This, however, has not hindered the community interest as pattern descriptions continue to proliferate. Dearden and Finlay’s (2006) review of pattern and patterns languages in HCI conclude with an agenda—requiring the HCI community to divert focus away from producing patterns to evaluating patterns in design.

**Motivation & Contributions**

Chung et al. (2004), Saponas et. al (2006), and Wania (2008) present controlled studies evaluating the impact of patterns on the quality of the designed product. While evidence of patterns impacting quality, from these studies, remains inconclusive, it is maintained that patterns may still be valuable as a communication tool. The HCI community appears to have adopted the basic pattern structure (i.e., problem, solution, and context) from architecture as a de-facto standard for documenting design advice. The effectiveness of this pattern structure is generally assumed, and largely, unquestioned for communicating interaction design advice. None of the prior studies clearly discuss how patterns might assist communication. The proposed research focuses on one specific aspect of design communication—interaction design advice. It explores the effectiveness of a pattern structure on interaction design advice produced. In doing so it explores the value of using patterns. Studying the impact of a pattern structure warrants further study because several of the promises made about patterns seem to hinge on the pattern structure.
Discussions on tool support for managing and using patterns are moving forward based on the assumption that this structure adds value, which also raises the question—are we building these tools on proven foundation (i.e., pattern structure)? It can be argued that the concept of structured format for documenting advice is not novel. Many such formats have already been proposed to document design decisions along with rationale (e.g., Claims). So what makes the pattern structure special or useful?

The purpose of this empirical study is to understand the value of capturing interaction design advice using a pattern structure. It will investigate effectiveness of the pattern structure for communicating interaction design advice. The research questions motivating the proposed study are as follows:

- Is there value in using a pattern structure to document interaction design advice?
- Are interaction patterns an effective format for capturing and communicating interaction design advice?
- To what extent does a pattern-like structure impact the overall quality of design advice? (Quality includes the following four metrics: time, context of use captured, usability relevance, rationale expresses, and overall quality)

Like in the case of design rationale, the advice captured may be more useful to people other than the authors themselves (Grudin, 1994). Molich et al. (2007) argue for the concreteness and level of detail present in the design advice as an indicator of usable design advice, i.e., some one reviewing this advice should have concrete suggestions to work on. Another indicator, according to them, is the usefulness of the design advice, i.e., the advice should help address a usability related problem by providing an effective solution. Jeffries (1994) requires that recommendations should include the trade-offs associated with the solution since no solution is a perfect solution. And there are always consequences associated any design choice. Based on these stated parameters, this study evaluates the effectiveness of the pattern structure by judging the impact of this structure on the effectiveness of the design advice. More specifically, the study
explores the impact of the structure of advice on the context of use captured, rationale expressed, usefulness, overall quality, and the time required to prepare advice.

The discussion chapter, which follows, traces the adoption of patterns by the HCI community and its roots. It documents the motivations discussed in the literature about “why patterns,” and summarizes the broad challenges encountered by the interaction pattern community. We discuss in more detail the empirical work conducted on patterns and interaction design to highlight the inconclusiveness of the findings, which sets the scope for the presented study around structure of advice. Based on the findings from this dissertation, we argue for cons and trade-offs amongst design issues as key elements for capturing and communicating design advice. We venture an enhanced structure for communicating advice as a way to operationalize our final recommendations.
CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This section discusses the relevant literature on patterns in HCI. It covers the following topics:

- Design Methodology and HCI
- Pattern Construct in Architecture
- Pattern Construct in HCI
- Pattern and Its Roots
- Promise of Patterns for HCI
- Problems or Challenges for Patterns in HCI
- Empirical Work on Patterns in HCI
- Rationale for Proposed Research

2.2 Design Methodology and HCI

Rittel (1984) explains problems in urban planning as 'wicked problems.' Expertise to solve a wicked problem is distributed, i.e., no one person can claim to possess the knowledge to solve it. Rittel calls this as a "symmetry of ignorance" problem. When multiple stakeholders with different expertise or experiences are involved in design projects, understanding what each group has to say, or understanding each other, may be a catalyst for successful outcomes.

Designing interactive systems in human-computer interaction (HCI) is not very different from solving "wicked" design problems. There is no one set definition for HCI, but the following one from SIGCHI helps set the scope:
Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.

Designing interactive systems requires a concerted effort on the part of designers, users, and other stakeholders. While we may exert some control over the design of the artifact, the true test of good design remains the fit between the designed artifact and the context of use (Alexander, 1964). So, some of the objectives for design methods in any discipline, e.g. engineering, urban planning, architecture, is to suggest methods and techniques to cope with the complexity of the problem space. The design methods aim to do this by providing necessary structure and guidance that may help stakeholders negotiate this 'fit'.

Pemberton (2000) asserts that it is a challenge for any design discipline to make design experience, about the process or product, explicit and usable to others. So, how do we share this accrued design knowledge about designing interactive systems in HCI? Over the years the HCI community has strived to make explicit what we know about designing interactive systems in various ways. This include principles derived from studies of human cognition for understanding the capacities and limitations of end-users (Card et al. 1984; Norman, 1986); basic tenets for an user-centered design process (Gould and Lewis, 1985); design lifecycle models (Mayhew 1999, Nielsen 1994); and also specific methods for evaluating usability of interactive systems (Nielsen 1994; Polson and Lewis 1990).

2.3 Pattern Construct In Architecture

In explaining a pattern in architecture, Alexander argued that each living space is designed to support a pattern of activities. For example, a family room or a dining room inside a

---

1 Cross (1984) provides a good discussion on different perspectives on design methodology from various design disciplines
house implies a certain purpose for that room. But, according to Alexander et al. (1979), not all rooms are alike; there are some rooms that make us feel more comfortable or are well-liked compared to others. Assuming that we have experienced such situations, e.g., pleasing family rooms, Alexander et al. (1979) suggest that it is possible to identify a spatial arrangement of elements in that room or a "configuration" that contributes to our positive experience. Furthermore, it is possible to tease out this particular arrangement or configuration in all family rooms that appeal to our senses. And this degree of agreement, about what makes us feel good about a space, may help us to identify an invariant that manifests itself in all the different preferred family rooms.

For the sake of readability, in rest of the discussion, citing Alexander (1977, 1979) would imply Alexander et al. (1977, 1979).

According to Alexander, a pattern describes a spatial configuration of physical elements in a space, which resolves a system of forces in that context. The system of forces refers to some conflicting requirements, or constraints present, e.g., family rooms need to be designed to support a sense of belonging or community, but at the same time allow just enough privacy to pursue individual hobbies and tastes without sacrificing the communal feeling completely. Alexander (1977) suggests an ALCOVE pattern (p.235) as a solution to the design problem stated earlier. This pattern is described in more detail under the discussion on a pattern format. ALCOVE pattern is one among 253 such patterns identified by Alexander, published as “A Pattern Language.”

Alexander (1977) define a pattern in architecture as

Each pattern describes a problem, which occurs over and over again in our environment, an then describes the core of the solution to that problem, in such a way that you use this solution a million times over, without ever doing it the same way twice. (p. X)
By referring to patterns as representing the core of the solution, or an invariant, Alexander suggests that the design advice provided requires further adaptation or contextualization, before it can be used. This adaptation is what makes each instance unique, yet similar in essence. The following description of the ALCOVE pattern would clarify this point further.

Alexander (1977) proposed a set format for describing a pattern. It begins with a reference number for the pattern and a concise pattern name; the name is supposed to communicate the about-ness of the pattern, e.g. LIGHT ON TWO SIDES OF THE ROOM (159) discusses designing a room such that light comes in from two sides of the room, either adjacent walls or opposing walls. The pattern name is followed by an image, usually a photograph showing how the pattern is instantiated in the real world, along with a description of the context. The context captures when and where we encounter this pattern, e.g., in the ALCOVE pattern, the context is family rooms, communal rooms, or any such spaces where a group of more than few people get together. Following this, a three-diamond marking signifies the start of the problem description (type-setting detail), and begins with a succinct definition of the problem. In the case of the ALCOVE pattern, the problem is stated as,

No homogeneous room, of homogeneous height, can serve a group of people well. To give a group chance to be together, as a group, a room must also give them the chance to be alone, in one's and two's in the same space (p. 829, Alexander et al., 1977).

This is followed by a detailed explanation justifying the basis of the problem, and when and where we might encounter this problem. For the ALCOVE pattern the context is community or family rooms, and the problem can be restated as the need for balancing the opposing requirements of privacy while retaining a feeling of community. Based on empirical work Alexander believes to be relevant, and examples of already build environments, he begins to
outline the properties desired of the solution. This outlining process is in fact a restatement or reconfiguration of the problem such that the solution becomes more apparent. This solution is stated in the in terms of a spatial arrangement or configuration, which describes how the space should be built or utilized. Here the solution for balancing the opposing forces in community rooms is stated as,

Make small places at the edge of any common room, usually no more than 6 feet wide and 3 to 6 feet deep and possibly smaller. These alcoves should be large enough for two people to sit, chat, or play and sometimes large enough to contain a desk or a table (p. 832, Alexander et al., 1977).

While the dimensions appear constraining, a moment of reflection may reveal that considering dimensions of greater than 6 feet in width would depend on how big the family room is, which highlights the role played by the context. Another three diamond marking, similar to the one found earlier, marks the end of the pattern description. The next section point readers to other related patterns than would help build and refine the pattern under scrutiny. The discovery of the ALCOVE pattern outlined here captures one of the ways Alexander (1979) discusses a pattern discovery process in architecture.

2.4 Pattern Construct In HCI

The pattern concept in architecture has since spawned several notable adaptations. Software engineering community was the first to notice a value of patterns. Gang of Four (Gamma et al. 1995), are be attributed for first adapting this concept in the form of design patterns. The increasing interest in “A Pattern Language” from the computer science community appears to coincide with the publication of software design patterns, which incidentally was published in 1994 (refer figure 2-1). Since then, the pattern concept has been repurposed by several other disciplines, e.g., the Pedagogical Pattern Project (Fincher, 1990a) and Patterns for Computer
Mediated Communication (Lukosch & Schummer, 2006). The emphasis of design patterns has been on reuse of object-oriented code or micro-architecture. Like patterns in architecture, design patterns capture a recurring problem and solution pair along with the context. It is documented in a consistent format, along with pseudo-code showing how to implement the solution; design patterns are generalizable to any object-oriented programming environment. GoF state the purpose as, "Design patterns help the designer to get a design right faster" (p.2). Dearden and Finlay (2006) provide a good chronological account on how patterns describing interface related problems branched away from discussion on design patterns at Pattern Languages of Programming (PLoP) conferences. Design patterns continue to be actively researched at PLoP conferences that take place annually.

This research is more concerned with discussions on interaction patterns, since this is the domain of proposed research. Unlike design patterns, which have a clear goal for reuse, the HCI community envisions a greater impact and role for interaction patterns. They propose interaction patterns may serve as a common language for users and designers to voice their ideas, and as a possible format to communicate interaction design knowledge to those interested in HCI (Erickson, 2000; Bayle et al., 1997; Borchers, 2000). These promises of interaction patterns are reviewed in greater detail in a later section (refer sections 2.6).

There is no one formal definition for a pattern in HCI. Realizing this, a CHI 2000 workshop on interaction patterns proposed the following definition (Borchers, 2000a):

An HCI design pattern captures the essence of a successful solution to a recurring usability problem in interactive systems. It consists of the components name, ranking, sensitizing example, context, problem statement, evidence (rationale, examples), solution, sketch, references to other patterns, synopsis, and credits.
While design patterns in software engineering were the first to adapt the pattern concept for capturing best practices in object oriented design, the HCI community has argued that the designed artifacts in the interaction design have a greater potential to affect the social and personal behavior of users (Fincher 2000) than object-oriented code. This is suggested as a reason why patterns in HCI should source their inspiration from patterns in architecture instead of design patterns. Tidwell (2005) too draws an analogy with architecture stating,

… 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.

A more concise definition for a pattern suggested by Tidwell highlights the core elements of a pattern,

A pattern describes possible good solutions to a common design problem within a certain context, by describing the invariant qualities of all those solutions.

In essence, interaction patterns follow a format similar to architecture patterns. It consists of a narrative (design guidance), structured around the configuration of interface elements that resolve a set of forces in a specified context. A pattern in architecture comprises of three elements: a description of the recurring design problem, a solution (and diagram) showing a configuration that brings into balance the “forces,” and a definition of the context in which the problem-solution pair resides (Alexander, 1979).

For example, a Location breadcrumb is an example of an interaction pattern. In the case of a “location Breadcrumb” pattern, van Welie (n.d.) states the interaction design problem faced as:
The users need to know where they are in a hierarchical structure and navigate back to higher levels in the (information) hierarchy.

He defines the context where and when this problem crops up as:

Sites with a large hierarchical information structure, typically more than 3 levels deep. Such sites are medium to large sized and include E-commerce Site, catalogs, Portal Site, Corporate Site etc. The site has got some type of Main Navigation that allows users to traverse the hierarchy. Users may want to jump several steps back instead of following the hierarchy. Users may be unfamiliar with the hierarchical structure of the information.

Based on proven solutions found on similar web-pages or sites, he suggests the following as the essence of the solution to address the problem:

The path shows the location of the current page in the total information structure. Each level of the hierarchy is labeled and functions as a link to that level. The current page is marked in order to give the users feedback about where they are now and should not be a link. Don't use the current page name in the breadcrumb as the only way to show section title, add a title anyway.

Home > Next Level > Next Level > **Current Page**

The path shows that a top-down path is traversed by using appropriate separators such as > or \ that suggest a downward motion. If the path becomes too long to fit in the designated place, some of the steps can be replaced by an ellipsis e.g. ‘...’. The path is placed in a separate ‘bar’ that preferably spans the entire width of the content area. It is placed close to the content area, preferably above the content area but below the page header.

van Welie also adds a rationale behind why this may be a good solution in this given context, where he succinctly explains how a breadcrumb is different from the primary navigation on a website:
The breadcrumbs show the users where they are and how the information is structured. Because users see the way the hierarchy is structured they can learn it more easily. By making each label a link, the users can quickly browse up the hierarchy. They take up minimal space on the page and leave most of the space for the real content. Breadcrumbs are not for primary navigation and should always be used together with a form of Main Navigation. Usability testing has shown that breadcrumbs are never cause trouble and that at least some people use them. So it is nearly always good to use them.

The name breadcrumb refers to the fairy-tale of Hansel and Gretel where a breadcrumb trail is used to mark the places Hansel has been. If the analogy were correct a breadcrumb should show the history of the users' actions rather than the position in the hierarchy. So the name breadcrumb is actually wrong...

In summary, in large hierarchical information structures like some web-sites (context) Location bread-crumb shown as Home>Level 1>Level 1.1>, clearly displayed to the user in a dedicated location on the page (configuration), would prevent users from getting lost in the levels while giving guidance to return to the place they started (forces). Unlike a pattern in architecture, in van Welie’s version of patterns, the rationale follows the solution. This appears to be more of a preference as Tidwell in her patterns explains “why” it is a good solution before describing “how” to implement the solution.

2.5 Reaction to Patterns
This section provides an overview of how the pattern concept was introduced to architecture and its later adaptations by the software engineering community and other disciplines, including interaction design. It presents a brief review of the how patterns and a pattern language is viewed within the architecture community. The discussion following this attempts an explanation as to why discussions on patterns persist in other communities of practice, despite the criticism.

Alexander (1977) is attributed for introducing the pattern concept to architecture. They believe patterns would provide a vocabulary and a language, a pattern language, that would
enable architects and non-architects communicate. According to them patterns are derived from practice, as in already built-up environment, not theory. Alexander et al. (1977) compiled 253 such patterns, which they published as "A Pattern Language" for architecture. Some of the arguments justifying patterns are rooted in an earlier work by Christopher Alexander published as "Notes of the Synthesis of Form." In a multi-part series, spanning three volumes: The Timeless Way of Building, A Pattern Language, The Oregon Experiment, Alexander and his colleagues outline the philosophy behind the pattern approach, explain ways in which to discover patterns present around us, show a way to document these patterns to make it shareable, and show how these patterns could form a possible language for design.

Alexander (1964) believes that the knowledge on how to solve design problems (in architecture) is already available to each one of us. He gives an example of 'primitive' cultures, or as he calls 'unselfconscious' cultures, where there are no 'architects' or 'designers'. The knowledge of how to design a house is tacit to each individual in that community (and culture), and there is one known way of building a house (vernacular architecture). As Alexander (1964) reasons, this knowledge is not taught using design principles or theory, but rather, passed on across generations, and gained through the actual practice of building. Such houses are built using materials that are native to that culture and suited for that environment (architecture without architects, Rudofsky, 1964).

But, in modern societies, or as he refers to as 'self-conscious' cultures, there are clearly defined roles of 'architects' and 'designers.' According to him, in self-conscious cultures, the knowledge about design seem to be imparted through the use of design rules and principles, more than the practice of building. This might be truer of software, or interaction design, than architecture or fine arts, which still employ an apprenticeship model. In interaction design, we can consider situations where we try and teach the ‘basic principles’ of good design, before
moving on to building a system/prototype. Here a classroom setting takes importance over a studio model. The reasons might stem from the infeasibility of adopting a studio model in circumstances where we may be trying to multiskilling new designers, and studios may not be able to produce enough designers to meet the market demand for new designers. According to Alexander, people have slowly distanced themselves from this knowledge of building, and materials, and have begun relying on architects to do the job. Individuals have lost confidence in their tacit abilities to design living spaces, and as a result, the designed spaces tend to reflect the designers' vision more than the inhabitants need. It begins to serve as an outlet for the architects' or designers' individuality. For example, homes designed by the acclaimed American architect, Frank Lloyd Wright, have a signature look commonly associated with Wright, as his style, and may not be representative of the inhabitants.

"A Pattern Language" effort was proposed in response to this taught helplessness in modern society. It was proposed as a way to explicate the experiential design knowledge of how to design living structures, and make it shareable. A "pattern" was chosen as a format to encapsulate this design knowledge in a way that could be understood and used by architects and non-architects alike.

2.5.1 Pattern Skeptics Within the Architecture Community

The concept of patterns, while applauded by some within the architecture community, for its departure from a conventional way of practicing architecture, it has also met with its share of criticism. Critics target an authoritarian style in which the patterns are authored. They also do not seem to agree with the proposition that “A Pattern Language” is sufficient to generate designs (spaces) of great variety or that the patterns have been empirically qualified. Moreover, they find the utopian ideas communicated in “A Pattern Language” detached from how present day society lives.
Saunders (2002) in his review of “A Pattern Language” highlights some of the contradictions present in the book; he believes Alexander et al. (1977) propose patterns as mere design ideas to consider, which imposes nothing on the reader, but the authoritative language indicates the contrary. Pattern description start sound more like a rule, i.e., there is one way to implement this pattern and no other.

According to Saunders (2002), taken literally “A Pattern Language” presents 253 rules, which dictate not only how to design, but how to live Alexander’s way. Saunders believes that the patterns discussed by Alexander reflect his personal experience and tastes than something universal. He believes this is a result of Alexander living and being exposed to certain specific cultures, mainly the western world. Protzen in his essay refers to this as the “poverty of a pattern language”, arguing that this language disregards the diversity that exist inside communities. “A Pattern Language” alone cannot account for this diversity in design of spaces.

What critics also find disturbing is a duality present in Alexander’s proposition that patterns can be proven empirically. On the one hand Alexander cites related empirical work, which according to him, provides a scientific foundation for the patterns. On the other hand, Alexander shuns quantifiable measures for defining quality of designs, finding support in what he calls a “Quality without a Name” that cannot be measured but felt (Dovey, 1990). Alexander claims that purpose behind patterns is to explicate a spatial arrangement that hopes to achieve this “Quality without a name.” The critics feel that if it cannot be measured, how do we know we have achieved it? Kohn (2002) and Dovey (1990) both point out that Alexander’s patterns have never been tested, to see whether it leads to a better design. This according to them is a weakness behind a pattern-based approach. It is argued that while some of the patterns, (e.g., LIGHT ON TWO SIDES OF EVERY ROOM (159)), seem to appeal to our commonsense about what might
be good, even such claims can be falsified in places where extreme temperatures may demand cooler rooms with no windows (Protzen, 1980).

These aspects of “A Pattern Language” paired with the proposition by Alexander that users of such a language can build homes or towns without the help of architects or planners, may make the architectural community uneasy. They do not see an average person picking up “A Pattern Language,” and making the transition into the role of a designer. Taken literally, users fulfilling the role as a designer may appear to be a tall claim. Also, in a discipline where individuality and novelty are prized, suggesting that anyone can design a building using patterns abstracted from built up environments may not be received well. But, if this view is softened to see users as being able to articulate their requirements from a space in spatial terms, as suggestions, to be shared with a designer/architect, it may sound more realistic. Participatory design and the Scandinavian tradition have shown value in involving users in the design process, not as designers per se, but as co-designers (Muller & Kuhn, 1993).

2.5.2 Interpreting & Adopting Pattern Concept By Other Design Communities

While such criticisms surround patterns in architecture, there is rising interest from other design communities, e.g. software engineering, human-computer interaction. Critics in architecture appear to be united in disagreeing with the design advice present in a pattern, the way it is written, and what “A Pattern Language” aims to achieve; the authoritative language used and lack of empirical work are some of the issues highlighted earlier. Nevertheless, some of the critics acknowledge “A Pattern Language” as probably the “most informative book on architecture” containing some very practical ideas on design (Kohn 2002). Saunders (2002) believes that if the reader disregards the suggested rigidity in the design advice present, i.e. not take the advice literally, they may across design ideas that consider and share valuable insights about how, “…
people have opposite needs that should both be satisfied in building design, needs for being public and private, with others and alone …” (Saunders 2002).

“A Pattern Language” continues to be on the Amazon.com best seller list for books in architecture, suggesting it continues to attract sufficient interest. Figure 2-1 shows a history of the number of citations received by “A Pattern Language” from the architecture and computer science community. Since 1994, the number of citations from computer science has surged ahead of architecture. If number of citations were to be used as a rough measure of interest, it would appear that “A Pattern Language” continues to draw more attention from Computer Science compared to architecture². Citation analysis is a common technique used in the information science to determine influential pieces of works (White & Griffith, 1981; White & McCain, 1998).

² It should be noted that a citation history does not reveal the nature of citation, or the context.
Figure 2–1. Graph showing the trend for citations received by Alexander et al. (1977) from (1972-2007) computed from science citation index (Web of Knowledge)

What divides the architecture community (i.e., criticism) in their acceptance of patterns does not appear to affect the other disciplines that subsequently adopted patterns. May be because the different disciplines do not share any artifacts in common with architecture. Alexander (1977) had proposed that patterns would serve building blocks of a language that would be used by architects and non-architects to communicate; a pattern structure was argued as a format that would enable us to make this design knowledge about architecture explicit, and in the process, shareable. It can be argued that software engineering would gain little from the advice present in designing living spaces (e.g., alcoves), but these are the very artifacts architects may agree or disagree with.

What unites and common to pattern literature in different disciplines (e.g., software engineering, human-computer interaction) appears to be an implicit agreement on the basic structure or format of the design advice: problem, solution, context, and a diagram. The benefits of a diagram may be more relevant to HCI than other disciplines although design patterns also use a UML notation to visualize the design advice. As Rank, Coill, Boldyreff and Doughty (2004) point out that compared to architecture, the pattern movement has met with little resistance in software engineering. The direction adopted in these new disciplines seems geared towards extending, formalizing and implementing the pattern concept.
2.6 Promise of Patterns for HCI

Interaction patterns are a hot topic in HCI! Over the past few decades the HCI community has pursued their interest in interaction patterns almost independent of design patterns movement (PloP conferences). While the software engineering community cites the primary goal of design patterns as reuse of best practices in object oriented design, HCI view patterns fulfilling
a larger purpose: as lingua francas for interaction design, as a way to capture and communicate
design experience, to educate novice designers and build organizational memory, and as a way to
express design rationale. Consequently, several conferences and workshops have sought to
explore the role of interaction patterns in design and evaluation of interactive systems (Bayle,
1997; Borchers, 2000). Patterns in HCI, although late on the scene compared to design patterns,
have now surpassed other design disciplines in the number of documented interaction patterns
(Henninger & Correa, 2007). This section reviews the promise of patterns for HCI.

2.6.1 Patterns as Lingua Franca

The landscape of interactive systems is fast changing. As interactive systems become
pervasive, we are seeing an increased role for systems in multiple contexts of use. Add to that,
disciplines involved in HCI are not restricted to cognitive psychology and computer science, but
also other disciplines, for example, anthropology, ergonomics. Given this diverse make-up,
“symmetry of ignorance,” or the notion of distributed expertise is apparent during the design
activity. Such aspects of HCI demand a common ground between the diverse disciplines, and a
need for a common language to exchange ideas across disciplinary boundaries.

Alexander (1997) suggested “A Pattern Language” as a way for architects and non-
architects to communicate. This remains a key motivation within the HCI for interaction patterns
too. Patterns are seen as a format to document, and communicate, interaction design advice.
Bayle et al. (1997) proposed pattern languages in HCI would serve as this common language or
design; as a way to address the growing diversity of design stakeholders and other disciplines.
But Bayle et al. recognized the lack of documented interaction patterns to fulfill this problem
then. They suggested that HCI as a community find ways to discover and document patterns that
might be useful in interaction design. Erickson (2000) makes the case for a common language
between designers and users, to bolster the communicative aspects of design. He believes that
pattern languages echoing Alexander’s proposal are the answer. He asks that such a language for
design be accessible to all stakeholders. Although not a discussion on interaction patterns per se,
he describes how conducting discussion on re-designing the town around concrete objects, and
creating transparency in the community mind-set, in terms of aggregate individual preferences
and values, helped create a common language made available to the residents. He illustrates the
benefits of lingua francas with the help of a case study on urban re-design of the city of Manteo,
NC. Patterns, according to Alexander, draw their power from practice, and not theory. They
represent the essence of proven solutions to a recurring design problem, and are situated in a
context of use. The concreteness present in the design guidance in patterns was suggested as one
potential reason for success with diverse stakeholders (Erickson, 2000).

Borchers (2000) cites

… the goals of an HCI pattern language are to share successful HCI design
solutions among our colleagues, and to provide a common language for HCI
design to anyone involved in the design, development, evaluation or use of
interactive systems (p. X).

This goal is echoed in another one of his claims that patterns need to be readable to users
to promote user participation in the design process; in other words, users should be able to
understand the guidance present in interaction patterns.

Having a common language for design is also seen as way to improve user participation
(Bayle, 1997,). Borchers (2000b, 2000c) feels that viewing patterns only as a form of expert
design communication neglects its true potential. Providing additional ways for design
stakeholders to express their ideas would offer a way out of the consultative or evaluative roles
they may be unintentionally cast into. Such benefits cannot be overlooked by HCI when
interactive systems are pervading multiple contexts of use (Bødker, 2006).
2.6.2 Patterns for Capturing and Communicating Design Experience

Given the broad goal of working towards lingua franca for interaction design, interaction patterns have been suggested as a format for capturing and communicating interaction design knowledge. Pemberton (2000) points out the challenge associated with sharing design experience.

Design has been argued as a science (Simon, 1983) and also as a craft (Alexander, 1978, Schön, 1987). This is especially true of HCI, where experiments in cognitive psychology have yielded principles that have value during design (Atwood, McCain & Williams, 2002), for example, Fitt’s Law. But at same time there may not be any rule that could predict how much the system requirements would change over time. At times like this, HCI relies on iterative build and test prototypes to gauge requirements, which seem to resemble the process followed by design studios or industrial design practiced (Dreyfuss, 2003). This may indicate that there is room for learning from design practice.

Similarly, patterns in HCI are supposed be discovered from proven solutions in practice, and not theory. It is not surprising to see expert designers use rule-of-thumbs or heuristics to guide their design process. These heuristics are not necessarily taught to them, but rather abstracted from design experience and assimilated into their design repertoire. Studies on expertise have shown the presence of such guiding principles as a way to differentiate experts from novice, advanced or even competent designers (Dorst, 2007; Hoffman et al., 1995). Heuristics, or rules-of-thumb, are also used in HCI. They have been used as a vehicle to deliver knowledge gained from practice (Nielsen, 1994), and continue to be used in the practitioner community.

While such heuristics have been shown to be useful in identifying usability problems (Nielsen, 1994), and addressing which may yield a more usable system, it is unclear whether presenting these rules-of-thumbs or heuristics alone would be sufficient. Question remains
whether such guiding principles used by experts are rich enough or even useful without experts themselves? Experts have the luxury of associating these principles with their design experience, but novice designers or users may find them of limited use since they often lack the experience. The onus and success of selecting and applying relevant guidelines in practice, seems to depend on the skill and experience of the designer. The specificity of design advice in style guides, and the generality of advice present in guidelines, may demand more of the designer (Mahemoff and Johnston, 1998).

Griffiths and Pemberton (2000, 2001) assert that, unlike heuristics, patterns are written in an engaging manner, and with wholeness present. They believe patterns are “conceptual tools” for design, but people still need to exercise creativity in applying the patterns. Dearden and Finlay (2006) summarize the potential of patterns as being more general than style guides, but more concrete than guidelines. Patterns attempt to communicate design advice at different levels of abstraction, larger system-wide issues and also finer details (interaction controls). While a pattern tends to be a richer format than guidelines, it may not capture every aspects of design experience or context. It would be impossible to document everything about the design experience, even if that is the explicit intent of capture. Nevertheless, patterns are said to possess the potential to be an improvement over generalized forms of design advice (e.g. heuristics).

2.6.3 Patterns to Enable Knowledge Re-use

An overarching goal for the HCI community for interaction patterns is to work towards a common language for interaction design, and interaction patterns have been suggested as a format to communicate design experience instead of guidelines (Bayle, 1997; Erickson, 2000; Pemberton, 2000). It is also assumed that having a resource comprising of proven solutions might be useful for accelerating the design of interfaces, and also for training novice designers.
Although HCI community does not view interaction patterns solely for re-use of interaction design solutions (Dearden and Finlay, 2006), such an expectation is not far behind. Granuland et al. (2000) look upon HCI design patterns as an inventory of solutions to commonly encountered UI design problems. Regarding the purpose of such collections, Tidwell (2005) believes patterns would allow inexperienced designers to develop confident and skillful designs without spending a significant amount to time “learning the hard way”. van Welie (2000) refers to his collection as “a toolkit for designing user experiences”. He suggests that interaction patterns available in his collection represent best practices in interaction design.

Sutcliffe (2000) argues that reuse of HCI knowledge is hindered by the way in which it is presented. It needs to be presented in a manner understandable to designers who may not have a background in cognitive psychology. While interface-design toolkits or style guides allow for some level of reuse, it does not require or allow any form of intervention or adaptation. This form of reuse is confined to either using interface widgets or rigid rules for interface style. Borchers and Thomas (2001) warn the community against a “cookbook” or “recipe book” approach to capturing HCI design knowledge for reuse since this may not be appropriate for a fast changing landscape of interaction design; he believes that any hard-coded widgets may quickly be rendered obsolete, wasting efforts spent in codifying it. Welie’s patterns in interaction design, and Tidwell’s pattern collections are some examples of pattern collections that aim to support reuse. Yahoo Pattern Library could be cited as an example of a pattern collection created within an organization as organizational memory.

While using past HCI design knowledge by designers for building interactive systems is one aspect of reuse, the HCI community has also proposed patterns for teaching novice designers based on successful solutions in the past. This approach is not that radical if we consider business schools teaching novice managers strategy and planning through concrete case studies.
Concreteness of design guidance through the use of examples of systems in use has been pointed out as an advantage of using patterns (Dearden and Finlay, 2006). Pemberton refers to patterns as a “knowledge representation formalism” to complement guidelines. He also suggests teaching patterns as a way to think about design; to enable students to abstract patterns from concrete examples of systems. Borchers (2002), Griffiths (2000, 2007), and Kotze et al. (2008) present arguments for introducing patterns in an educational setting to teach novice HCI principles to novice designers.

### 2.6.4 Patterns as Design Rationale

Moran and Carroll (1996) argue that the designed artifact is the product of a design process. This artifact represents the design choices made during the design process, but the assumptions, trade-offs, and negotiations may not be apparent from inspecting the artifact alone. According Moran and Carroll, design rationale captures not only the reasons behind the design, but is also generated for some specific purpose. They posit that capturing design rationale may prove valuable for communication, coordination, and memory about the design and the design process (i.e., knowing the rationale behind the design may help understanding between members of the design team, communicate the features to other design stakeholders, allow other designers to build on existing designs, and possibly serve as a roadmap of decision decisions to structure and coordinate design activities). Moran and Carroll (1996), in their edited volume, present a cross section of research on design rationale (DR) and DR notations (e.g., Procedural Hierarchy of Issues (PHI), Questions Options and Criteria (QOC), claims analysis).

Rationale plays an important role in the advice presented in patterns. Although Alexander (1979) discusses the three core elements of a pattern as context, system of forces, and configuration that bring the forces into balance, in the narrative describing the pattern, the rationale or why it is a good solution (configuration) is what attempts to persuade the reader. For
interaction patterns, both Tidwell and van Welie include a “why” section in their pattern
description. The rationale they express draws upon relevant cognitive principles or experience
gained from observing users interacting with interactive system. This rationale is the connective
tissue between the description of a interaction problem encountered, and a possible solution.
Pemberton (2000) argues that rationale may be essential for a deeper understanding of the
provided design advice, which may not be present in guidelines or style-guides. Dearden &
Finlay (2006) suggest that guidelines have some rationale present, but are more concise than
patterns. They continue that although patterns contain rationale, expressing rationale is not its
sole purpose. The rationale is presented to describe why the suggested solution is a good solution,
for a problem in a specific context, but always as part of design guidance.

Patterns are not a substitute for all other forms of DR. While the presence of rationale in
patterns shows overlap with other popular DR notations, there are also differences. Moran &
Carroll (1996) and Regli et al. (2002) assert that the way in which DR is captured invariably
depends on why it is being captured and how it might be used later. Accordingly, Moran and
Carroll classify DR as those produced as a by product of a design process, DR produced to
structure and represent the design space (options considered), or DR indexed by the design
feature. In this regard, patterns appear capture design rationale tied to a design feature, and not
the history of deliberations. Patterns are not about a specific design solution, but capture the
essence or the core of proven solutions in design practice.

There are comparisons of claims and patterns in previous literature (Dearden and Finaly,
2006). Dearden and Finlay point out that patterns differentiate themselves from claims by
including an explicit problem statement. Claims represent trade-offs, pros and cons, of
implementing a design feature within a specific usage context. Some have argued for claims as
psychological design rationale embedded in the designed artifact (Carroll, Singley, & Rosson,
1992; Carroll & Rosson, 1991; Singley & Carroll, 1996). Compared to patterns, claims are about a specific design feature, and in that sense the trade-offs are also tied to that specific feature. The example pattern shown in figure 2-3 shows the pros and cons of implementing a specific type of pop-up notification.

**Using MSN Style Pop-Up Alerts**

A clickable window about 1/24th the screen size appears in the bottom right hand of the users work area for a limited number of seconds, briefly alerting the users of a certain event, then disappearing. If the user clicks it, it will give more details about the alert and possible actions to take. (+) This feature would allow users to gain quick overview of the important and relevant information. (+) They can access more information, if they choose to, by clicking on notification. But, (-) could obscure or interfere with the primary task view.

Figure 2–3. Example of a claim describing the advantages and disadvantages of a “pop-up style alert” adapted from (A. Fabian et al., 2006)

In summary, Carroll (2006) refers to patterns (and PL) as a powerful, yet simple idea that could be explored. In his comments on the review by Dearden and Finlay (2006), he states

"… patterns address longstanding and fundamental questions about sharing and applying theory in HCI: They provide a moderate level of abstraction, concrete enough that users, application designers, and system architects all can contribute, understand, and share understanding. They help to anchor design rationale and design values, making it easier for designers to articulate and manage the embedded meanings and consequences of their designs" (p. 2).

This statement reiterates the promises about interaction patterns: lingua franca, format for capturing and sharing design knowledge, design reuse, and finally, design rationale. The next
section reviews some of the challenges faced by the HCI community that may prevent them from realizing these promises.

2.7 Problems or Challenges For Patterns In HCI

The section on the “promises” of interaction patterns summarized some of the motivations behind interaction patterns effort. Patterns as lingua franca for design was suggested early as a way to address the growing complexity and diversity in HCI (Bayle et al., 1998; Erickson, 2000). But, at the time, lack of documented interaction patterns prevented investigating this claim further. However, subsequent years witnessed a rise in the discovery and documentation interaction patterns. Tidwell’s and van Welie’s interaction pattern collections are results of some of the independent efforts. According to a survey on patterns available in different disciplines, Henninger & Corrêa (2007) report patterns in HCI lead other disciplines with over 400 documented patterns.

As number of interaction patterns proliferate, it signals new challenges for the HCI community to tackle, some of which include: agreeing on a common format for documenting interaction patterns; deciding on an organizing principle for managing a growing number of patterns; providing tool support to aggregate, search and use pattern collections; and demonstrating the value or empirical benefits of using interaction patterns for design.
Among these many challenges, demonstrating benefits of using patterns in design practice, e.g., value to the designed product or the design process, should take precedence over the other challenges. If we are unable to demonstrate this value, expending efforts to come up with more patterns or developing comprehensive tool support might yield little value. Such a situation may be analogous to spending money and time to build a software application that delivers no noticeable benefits to its users or the organization.
2.7.1 Common Format for Documenting Patterns in HCI

The interaction pattern format draws its inspiration from Alexander’s version, but there are differences in how different pattern authors have operationalized the pattern concept in HCI. According to Alexander (1979), at the heart of a pattern lies a description of the context, a system of forces, and a spatial configuration that brings these forces into balance. Table 2-1 shows similarities and differences in how the pattern concept is operationalized in HCI.

Table 2-1 Comparing some common pattern formats in HCI

<table>
<thead>
<tr>
<th>Tidwell</th>
<th>Welie</th>
<th>Yahoo</th>
<th>Van Duyne et al.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. What</td>
<td>1. Problem</td>
<td>1. Problem</td>
<td>2. Problem*</td>
</tr>
<tr>
<td>2. Use When</td>
<td>3. Use When</td>
<td>2. Use When</td>
<td>1. Background</td>
</tr>
<tr>
<td>4. How*</td>
<td>4. How*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Examples</td>
<td>Examples</td>
<td>Examples</td>
<td></td>
</tr>
<tr>
<td>6. Implementation</td>
<td>5. Accessibility</td>
<td>4. Consider these other Patterns</td>
<td></td>
</tr>
</tbody>
</table>

* A major part of pattern explanation in this section. The pattern sections in *italics* are unique to each pattern format.

It is easy to see how pattern collections and pattern authors the pattern description differently. Overall, the pattern description attempts to capture and explain similar things (e.g., context, problem, solution), each author refer to these sections using different terms. Welie, van Duyne et al. (2006), and Yahoo, all include an explicit problem statement, but Tidwell does not.
In Tidwell’s pattern format the WHAT section reads more like a summary of the solution. Table X compares the problem statement from Welie BREADCRUMBS Pattern with Tidwell’s version. Alexander’s problem statement for the ALCOVE pattern is included only for giving the readers an idea of how a problem statement sounds like in architecture.

Table 2-2 Comparing problem statements presented by pattern authors in HCI compared to Alexander’s (1977) version.

<table>
<thead>
<tr>
<th>Author</th>
<th>Problem Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tidwell</td>
<td>On each page in a hierarchy, show a map of all the parent pages, up to the main page. (p. 78, Tidwell, 2005)</td>
</tr>
<tr>
<td>van Welie</td>
<td>The users need to know where they are in a hierarchical structure and navigate back to higher levels in the hierarchy</td>
</tr>
<tr>
<td>Alexander</td>
<td>No homogeneous room, of homogeneous height, can serve a group of people well. To give a group chance to be together, as a group, a room must also give them the chance to be alone, in one's and two's in the same space” (p. 829, Alexander et al., 1977).</td>
</tr>
</tbody>
</table>

Another difference is where the authors include the bulk of their explanation, or justifications, for the pattern. A major portion of Alexander’s pattern description is presented after the problem statement and before presenting the final solution. Van Duyne et al. (2003) and Borchers (2000) both adopt this approach. Both Tidwell and van Welie, however, provide the major portion of their justifications under the WHY and HOW sections. Also Table 2-1 shows that these sections are organized in different sequence, e.g., Tidwell describes WHY it is a good solution before discussing HOW to implement this solutions, but van Welie choose to discuss WHY it is a good solution after suggesting how.

It could be argued that the section names and their organization may not cause much problem if we were to read a pattern in totality. While we can expect this from someone familiar with these variations, it might cause unnecessary confusion for someone unfamiliar with this
domain, or a newbie. Lack of an accepted common format could impede the usability of pattern collections. Seffah & Javahery, 2002; Dearden and Finlay, 2006; Segerstahl & Jokela, have pointed this out as a problem that needs to be addressed. Subsequent efforts to address this problem have resulted in a XML based Pattern Language Mark-Up Language (PLML), which provides a standardized machine-readable format for documenting patterns. This, however, does little to the presentation layer or what the reader sees, but provides flexibility to interpret and relate the different section when using tool support.

2.7.2 Organizing Principle for Interaction Patterns: Pattern Languages or Collections

A survey of available patterns in different disciplines show there are over 400 patterns available in HCI spread across different collections (Heninnger and Correa, 2007). But, while pattern descriptions proliferate, discussions on an organizing principle for managing this growing number of patterns remain unresolved. Sutcliffe (2001) argues that interaction patterns lacks a sound theory of abstraction, which prevents reuse and prohibits further contribution (more patterns) from the HCI community. Such a framework would allow us to locate the gaps in the collection or language, which can then be addressed.

There are different perspectives dictating solutions to organization of interaction patterns. On one hand, several multi-dimensional organization schemes have been proposed for managing interaction patterns (Borhcer, 1999; Fincher & Windsor, 2001; van Welie and van der Veer, 2000). On the other hand, there are those who are pushing for exploring pattern languages for organizing interaction patterns [21, 30].

Fincher & Windsor (2001) presented four attributes to qualify an organizing principle: Taxonomize, allow users to quickly locate patterns in a large collection; Proximate, zooming in and out, revealing more or less pattern neighbors, move between levels of abstraction; Evaluative, allowing designers to approach the problem from different viewpoints; and Generative, allowing
designers to envision and build new solutions, in other words, combine the interaction patterns in novel ways to yield a new whole. Another possible organization scheme suggested by Borchers’s [8] uses three dimensions to classify an interaction pattern: Level of Abstraction, Function, and Physical Dimension. Level of Abstraction, according to him, is the most important dimension. This would allow classification of interaction patterns from low-high level of abstraction: object-level < style level < task-level patterns. The second dimension of Function allows distinguishing interaction patterns that deal with perception, what users see when interacting with the interface; manipulation, how users would interact with the interface elements; and navigation, how users would navigate or move around using the interface. The third and final dimension, Physical Dimension, identifies interaction patterns that relate to spatial arrangement (space with respect to the screen), sequence of tasks, or with continuous time (animation).

In contrast to a collection approach, some in the HCI community have called for organizing patterns as languages. In architecture, a pattern and a pattern language are not independent constructs. On the contrary, a pattern does not exist in isolation. Alexander’s emphasis has been on building pattern languages and not on discovering patterns alone. He explains this relationship between patterns and a language Alexander [3] as

“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” (p. 313).

Alexander (1977) highlights this as an important role of pattern languages,

“The sequence of patterns is both a summary of the language, and at the same time, an index to the patterns” (p. xviii).

Salingaros (2000) asserts that a connected set of patterns provide a “framework on which any design can be anchored.” He goes on state that the relationships between patterns in a pattern
language constrain the number of design possibilities while still allowing a great deal of variations in design. According to him, the “narrowing of possibilities” is an essential aspect of any practical design method—a necessary feature found lacking in the different organization schemes. Todd, Kemp and Phillips (2004) argue for pattern language in HCI and propose three types of validation to determine if a networked structure of interaction patterns indeed represents a pattern language: Validity of individual patterns; internal validity of a pattern language; and external validity of a pattern language. Within the context of interaction design, a network of interaction patterns possess external validity by determining whether the patterns help achieve certain desired goals, e.g., design completeness, detail, a quality system (Todd, Kemp, Phillips, 2004). A pattern language is internally valid if it is truly generative; it is generative if the connectivity between patterns show how lower level patterns can be combined to build complex wholes, and vice-versa. Erickson (2001) believes that ignoring the generative aspects of a pattern language undermines some of potential benefits of patterns, i.e., “… allowing people to see and talk about the interdependencies in a particular design domain” (p. 226).

Despite these different perspectives, pattern collections continue to organize patterns based on functionality, e.g., Tidwell categorizes the BREADCRUMBS patterns under the category GETTING AROUND: NAVIGATION, SIGNPOSTS, and WAYFINDING (p. 5). There is little empirical evidence in the literature supporting organizing interaction patterns as collections or languages.

### 2.7.3 Tools to Aggregate, Search, and Use Patterns

The narrative structure of patterns makes the task of finding and evaluating patterns a challenge. Although this structure improves comprehension, the process for selecting patterns remains tedious. Due to the textual nature and the level of detail captured in each pattern, a user may get lost when searching and reviewing multiple patterns. This is compounded by the fact
there is no centralized repository for patterns (Gaffar, 2003) Some of the pattern collections currently available on the Internet, e.g., http://welie.com or http://designinginterfaces.com, are maintained by independent authors and are not complete. As a result, each collection follows an organizing principle imposed by their respective authors. A design exercise during CHI 2002 pattern workshop, using two of the popular collections, determined that existing collections do not provide any guidance for design, and prior knowledge with the collections or context of the design dictated success of the exercise. Because of the differences in how the patterns are authored, named, organized and related (Gaffar, 2005; Segerstähl 2006), it could cause the designers to inadvertently neglect patterns because it has been filed in a certain way. Conflicting or redundant pattern names could also cause the reviewer to miss reviewing a relevant pattern altogether. This makes the task of reviewing relevant patterns tiresome. Deng, Kemp and Todd (2005) argue that poor accessibility of patterns and Pattern languages may be one reason why we do not see reports on how patterns are used. The effort spent in finding patterns undermines its inherent benefit.

A tool-based approach has been proposed and discussed in the literature to help improve accessibility of patterns (Deng, Kemp & Todd, 2005). The CHI 2003 workshop on “HCI patterns: Concepts and Tools” brought together experts to demonstrate tools for accessing patterns. This workshop helped establish the specifications for pattern tool to manage disparate collections, and also suggested PLML as a standard for documenting interaction patterns.

Deng and Junhua (2005) categorize the requirements under three major functions:

- **Catalog**: The tool should be able to collect and present patterns, allowing users to view patterns that exist in the collections. In addition to this the tool should provide support for creation and submission of new patterns.
• Manage: management of patterns refers to the ability to modify and manipulate patterns and pattern collections. It includes being able to search for patterns based on multiple organization schemes and through the manipulation of forces. It should also support versioning and variants of existing patterns for e.g. a search pattern should be different for a novice and expert users but is still a search pattern.

• Pattern-based Design environment: This calls for a design environment where users could use the management features to pull together related patterns, create new relationships between patterns and in the process develop a pattern language for their individual design tasks. The tool needs to support the users to incrementally build the design, refining it using lower level patterns.

But, we are yet to see a tool that accomplishes all the three functions.

2.7.4 Evaluating Benefits of Interaction Patterns for Design of Interactive Systems

Research effort to address some of the problems stemming from non-standardized pattern formats, lack of an organizing principle, and need for tool support have seem to have captured the attention of the HCI community and gaining momentum. While exploring solutions to address the above-mentioned problems may be right direction, we are yet to see conclusive evidence that demonstrates the value of using interaction patterns.

Coming up with ways to standardize patterns, or spending efforts developing tool support to retrieve patterns, may not be worth much if are unable to show patterns as indeed useful. We are aware of the “promises” made about interaction patterns (e.g., lingua franca, as a format for communicating interaction design knowledge), but we are yet to see empirical evidence that suggest whether these promises are realized.
Dearden and Finlay’s (2006) review on pattern languages in HCI converge on similar questions. Their doubts stem from the lack of empirical work in the area of interaction patterns demonstrating value of patterns to the designed product or the design process. They applaud the interest and effort from the HCI community, which has led to several conferences, workshops, and pattern collections. But, while pattern descriptions proliferate, there has been little done to explore interaction patterns from an empirical perspective. Dearden & Finlay put forth a new research agenda for the HCI community where they prioritize four possible areas of inquiry:

- Exploring the value of using pattern languages in education and design.
- Determining ways to organize and apply patterns, that exist at different levels of abstraction, in a cohesive manner that address issues stemming from the social context of technology use down to the interaction level details?
- Expanding the scope of patterns such that these are not tied down to any particular interaction paradigm, and at same time explicating the process by which patterns can be discovered in a systematic manner to grow the body of interaction design knowledge through patterns.
- Making clear how design values, shared within the HCI community, are encapsulated and explicated in patterns.

This agenda raises some specific issues about demonstrating benefits of using patterns and pattern languages that need research attention if we are to realize some of the promises made for HCI. These issues also seem to resonate with the criticism received by “A Pattern Language” from the architecture community, where the empirical basis of patterns has been questioned (Dovey, 1990; Kohn, 2002; Protzen, 1980). The critics of “A Pattern Language” too seek evidence of the value of using patterns for design.
In HCI, we have begun to see empirical work; both in the form of controlled studies, and case studies, that try to investigate the value of using patterns for design and education.

2.7.5 Summary Of Challenges and Agenda For Proposed Work

This section reviewed some of the challenges faced by patterns research in HCI. The discussed challenges by no means represent an exhaustive list of open issues, but it attempts to capture some of the major themes or concerns raised in the literature. It mainly targets an inconsistent format for patterns, lack of an organizing principle for managing patterns, need for tool support, and finally, lack of empirical evidence demonstrating value of a pattern-based approach for designing or the design process.

The direction adopted by the HCI community, towards a consistent format and tool-support, appears to be the right one, but existing research seems to have deemphasized investigating the value of interaction patterns and pattern languages to communication. It is generally assumed that there are benefits to using patterns as a format for communicating interaction design advice. Carroll (2006) believes patterns have potential to make an impact in HCI, but needs more attention and work. Erickson (2000), responding to doubts raised about his “lingua franca” proposal that patterns have not been applied in real interaction design practice, believes that availability of patterns in HCI is an indication of serious work beginning in this area.

The next section discusses in greater detail empirical work on patterns and pattern languages in HCI and its contribution to interaction design our knowledge about interaction patterns.

2.8 Empirical Work on Patterns in HCI

We have begun to see empirical work in the area of patterns in HCI as a response to the concerns raised by members of the HCI community (Dearden and Finlay, 2006; Carroll, 2006). This section summarizes empirical studies conducted to evaluate the value of patterns in HCI.
These studies can be broadly classified into two broad categories: controlled studies, and case studies, as shown in figure 2-6. The following discussion would review case studies first, and the controlled studies follow.
2.8.1 Impact of Patterns in Education: Case Studies on Teaching With Patterns

Borchers (2000), based on his experience teaching two university level courses on interaction design, reported that HCI patterns leads to above-average retention of design principle. He believes patterns are useful tools for teaching HCI design principles. In the first study 26 undergraduate computer science (CS) students were given an introduction to patterns, its origins, and told that patterns are a way to capture HCI design concepts. They were asked to use patterns relevant to their project prototypes by Tidwell’s pattern collection, which they studied for fifteen minutes. As an evaluation of learning and retention, student feedback on patterns was obtained as a part of the course evaluation. This evaluation covered four aspects: remembering patterns, usefulness of patterns for communicating HCI concepts, usefulness to the prototyping exercise, and future benefits and use. Borchers reports that students rated usefulness of patterns as high, but
usefulness to the project was rated poor. In a second study conducted with a diverse audience of students (e.g., CS, education and learning technology), students were asked to author patterns of their own, after receiving introduction to topics in patterns covered in their textbook (Borchers, 2000). Over the duration of the course, the patterns were edited, and the final pattern was used as an indicator of success. In addition to this, general feedback received on course evaluation was suggested as an indirect measure of gauging student reactions. The overall score was slightly below than typical CS courses, which the author attributes to introducing unfamiliar concepts (patterns).

Griffiths and Pemberton (2000) suggested teaching HCI design principles by setting students practical exercises in using a pattern language in the process of design. They believe the ability to notice successful solutions as patterns is a more critical skill than familiarity with set of patterns of a pattern collection. Based on this assertion, a small group of 11 third year students of interaction design were asked to identify and document four patterns. All students received instruction on the pattern approach and directed to relevant readings. But, the authors discovered that students had trouble completing the exercise. To assist them, students were suggested to consider modifying existing guidelines and draw upon individual experiences. The authors informally report that students chose to rely on personal experiences, especially bad instances, to motivate their pattern discovery. According to the authors, of the thirty-seven patterns reported, only few achieved the level of abstraction desired of patterns. The rest seem to be more at the level of specific examples of bad design, and ways to address it. The authors argue that good design is hard to detect, which may have made it difficult to notice. Also, they believe that the students were not prepared enough in terms of skill to abstract patterns.

Kotze, Renaud, and Biljon (2008) report a study comparing the effectiveness of teaching HCI principles using anti-patterns. They hypothesized a difference in performance between
students taught using positive guidelines and those taught using negative guidelines. Anti-patterns by their definition represent negative patterns, i.e., capturing design practice that is destructive or yield a negative outcome. In this study, however, anti-patterns were represented by negative guidelines, which according to the authors capture the gist of the anti-pattern. They argue that since the objective is to study the efficacy of using negative practices to teach students, using a complete pattern description would be distracting. The participants were third year computer science students with at least three years of experience in software and interface design. Fifty-four students were assigned to two groups, a pattern group with 43 students and anti-pattern group with 11. Both groups received training on the use of color in interfaces, handling error reporting and data entry instructions, and Fitt’s law. As an intervention, pattern group were taught emphasizing positive advice, e.g. make sure buttons use bright colors; the anti-pattern group received training on the same topic, but the design advice emphasized what not to do, e.g. do not use low contrast colors for buttons. Students were asked to design a webpage asking them to use instructions given in class about the three topics. Four expert judges graded student designs, such that the final score received by each design was an average of four expert ratings. Based on comparison of group means, using the intervention as an independent variable and final grade/score as the dependent variable, the authors report students taught using patterns (positive guidelines) performed significantly better than those taught using anti-patterns (negative guidelines).

2.8.2 Impact of Patterns in Participatory Design: Case Studies Involving Users

Dearden, Finlay, Allgar and McManus [15, 16] focused on a participatory design process, using pattern languages as a way to involve the users. They believe, unlike software engineering, HCI should not restrict the use of patterns as a form of expert communication. This study involved designing paper prototypes of web applications with users in a facilitated session (expert present).
The users ranged in experience from no-experience to some experience in design of similar systems. They manipulated when the pattern language was introduced in the design process resulting in three configurations: two hours before the experiment, at the start of the experiment, or during the experiment. At all times, the experiment involved a facilitator and one user. The participants were told that the patterns are not compulsory or definitive. They conclude that patterns enabled novice users to design, while supporting expert designers. They also found the prototypes were coherent and the participants found examples present in patterns helpful. Overall, participants report a positive experience using patterns. The authors acknowledge that there was insufficient data to discuss quality, and the findings are based on the authors’ experiences conducting this study, and participant feedback.

2.8.3 Impact of Patterns on Quality of the Designed Product: Controlled Studies

Chung et al. (2004) explored whether pre-patterns were helpful for communication between designers, and whether using pre-patterns yielded better designs in the area of ubiquitous computing (Ubi-Comp). Pre-patterns by definition are not Interaction patterns, but represent candidate interaction patterns emerging in Ubi-Comp. This two-part controlled study compared use of pre-patterns by experienced and novice designers. Designer pairs were randomly assigned to one of the two conditions: patterns and no pattern group. The pairs were homogeneous, i.e., a high experience designer was paired with another high experience designer. The authors describe the high experience pair to have at least 8.5 years of design experience between them, and the low experience pair had at least 4.5 years of design experience. The study used a total of 9 pairs. Participants were emailed 45 pre-patterns two days before the experiment. In the first part of the study, participants were asked to use pre-patterns to complete a heuristic evaluation of a location-enhanced bus locator. Textual descriptions of usage scenarios and storyboards were made available to the design teams. This was followed by the second part, which was a design activity
to create a location-enhanced service for shopping mall patrons. Design teams were given 80
minutes to complete low-fidelity prototypes using paper, post-its or a whiteboard. They were then
asked to present their design to the authors as if pitching an idea to a client, which was
videotaped. Based on feedback received from the participants about usefulness of pre-patterns,
participants rated usefulness for future projects greater than design, and rated usefulness for
design more than for evaluation. The differences in ratings received for usefulness for each of
three activities, however, were not statistically significant. The designs created where evaluated
by three judges on a scale of 1 to 7 on three parameters: creativity, completeness, and quality.
Based on a comparison of mean ratings awarded by judges, the patterns group tended to have a
higher means score than the no patterns group. But, the high-experience control group scored
higher than high-experience with patterns group on quality. In a second follow-up study by
Chung et al. (2004), six professional designer pairs and six student designer pairs were assigned
to pattern and no pattern condition. This study reused the design task from the previous study, but
changed how participants were introduced to the pre-patterns. The numbers of patterns were
reduced from 45 to 30, but instead of emailing pre-patterns, participants were allowed 15 to
review the patterns followed by a ten-minute quiz. High experience designers report pre-patterns
to be more useful for the design activity than low experience designers.

Saponas, Prabhaker, Abowd and Landay [27] empirically evaluated the impact of pre-
patterns on early stage designs for digital home applications and communication of design ideas.
The authors suggest that pre-patterns had a ‘positive effect’ on early stage design, and would be
useful to designers working with unfamiliar domains. In this study, participants were all
professional designers with varying years of design experience. Forty-four designers were paired
up and assigned to pattern or no pattern group. The intervention attempted was exposure to a set
of pre-patterns, which was available to the pattern group. Each designer pair was asked to design
a low-fidelity prototype (interface) for a home food inventory system akin to a smart refrigerator that remembers consumption rate and recipes. The designs produced by each team were evaluated in two ways: heuristic evaluation and expert judging. The heuristics in this case were specific to the area of ubiquitous computing, and created by the authors targeting eight specific topics in this domain: attention, adoption, trust, conceptual models, interaction, interface, appeal, and application robustness. Three evaluators performed the heuristic evaluation. The evaluators found 220 unique issues for the control group, and 183 issues of the patterns group, after aggregating issues across the three evaluators and removing duplicates. The difference in the number of issues found was not significant. The expert judges evaluated three parameters of the design: completeness or breadth, detail or depth, and overall quality or is it a good solution? Here too, difference in the mean ratings received for completeness and quality was not significant. On a scale of 1-7 for completeness (i.e., very undetailed to very detailed), the control group received 3.7/7 compared to pattern group rating of 3.6/7. It was a similar story for quality ratings with control group rated at 3.6/7 and pattern group rated at 3.5/7. But, in the case of detail, the control group performed better than the patterns group, scoring 3.9/7 and 3.4/7 respectively. The difference in the mean rating was also statistically significant tested at p<0.1. The authors determined pre-patterns lead to higher quality designs on the basis of fewer heuristic violations.

Wania (2008) cites internal validity issues with studies presented by Chung et al.(2004) and Saponas et al. (2006), and highlights that these past studies were unable to show impact of patterns on the quality of the designs. Although these studies represent first two controlled studies presented in the area of patterns in HCI, Wania points out that pre-patterns may not be the same as patterns, a fact acknowledged by the authors of pre-patterns too. While patterns are derived from existing system, pre-patterns are candidate patterns, and are not prevalent in use yet. Another issue addressed by Wania is that the other two controlled studies do not compare pre-
patterns with any other design and evaluation technique in HCI. Dearden and Finlay (2006) suggest that such comparisons are necessary, and have been performed before (Gray & Saltzman, 1998; Jeffries et al., 1991; Wania, 2008). Responding to these concerns, Wania studies the impact of patterns on the design of interface retrieval systems. As a comparison, in addition to the control group, her study uses guidelines as the third group. Her participants were students at an information and technology school, who had completed at least one course in HCI. Participants were randomly assigned to one of three groups: patterns, guidelines, or control. The pattern group was exposed to an information retrieval (IR) pattern language (Wania and Atwood, 2007), and the guidelines group was exposed usability heuristics (Neilsen and Molich, 1994). Each participant was asked to sketch IR interfaces (paper prototypes) for completing a basic search and an advanced search. Two expert judges rated the finished prototypes for completeness, detail and overall quality on a scale of 1 to 7. The study reported that the patterns group performed better than the guidelines group and the control group. Also, the control group performed better than guidelines group. These differences in the mean ratings awarded were not statistically significant, but the pattern group had a higher number of patterns present in their designs, which was statistically significant compared to the other two groups. Wania concludes that patterns may not help early stage design, or the first iteration of the product, but she feels patterns may help in communicating interaction design knowledge.

Golden, John & Bass (2005) show that usability supporting architectural patterns (USAP) enabled software architects to perform better on a redesign task for implementing a cancel function for a given design. USAP documents usage scenarios that dictate changes to the software architecture and presents a solution to this problem (Bass et al., 2004). Golden, John & Bass test the usefulness of USAP with SE students. The USAP consisted of a usability scenario (U), general responsibilities(R) for involved entities, and a sample solution (SS). Although USAP are
not interaction patterns, it is still an empirical study on the format used for communicating design advice. USAP subsections consisting of a usability scenario, general responsibilities, and a sample solution roughly maps into a problem, context, and solution set-up. USAP also seem to bridge software architecture and HCI in some way since it deals with software architecture based solutions for usability related problems/scenarios. For the study the authors considered time on tasks and number of possible responsibilities identified (19 pre-identified responsibilities validated by experts). Participants were randomly assigned to one of three groups. Based on the group, participants received either U, U + R, or U + R + SS to test the explanatory power of USAP. They found time on task was not statistically significant, but the group who received the complete pattern (U + R + SS) did significantly better than just "U", indicating the value of the complete pattern and design guidance.

2.8.4 Summary Of Empirical Work

The case studies attempting to teach HCI design principles using patterns summarize report their findings as the following:

- HCI patterns lead to above-average retention of design principle (Borchers, 2002)
- Patterns are useful tools for teaching HCI design principles. (Borchers, 2002)
- HCI design principles should be taught by setting students practical exercises in using a pattern language in the process of design (Griffiths and Pemberton, 2000)
- The ability to notice successful solutions as patterns might be more useful skill than familiarity with set of patterns of a pattern collection (Griffiths and Pemberton, 2000)
- Students taught using patterns (positive guidelines) performed significantly better than those taught using anti-patterns (negative guidelines) (Kotze et al., 2008).

The case studies on using patterns in a participatory design imply the following:
• Patterns enabled novice users to design, while supporting expert designers.

• Enabled the user-designer team to come up with coherent designs

• Participants found examples present in patterns helpful.

The controlled studies suggest new information about using patterns in design as the following:

• Participants rated usefulness of pre-patterns more for future projects than for the design activity (Chung et al., 2004).

• Participants rated usefulness of pre-patterns higher for design than for evaluation (Chung et al.)

• High experience designers report pre-patterns to be more useful for the design activity than low experience designers (Chung et al.)

• Pre-patterns would be useful to designers working with unfamiliar domains (Saponas et al., 2006)

• Patterns may not help early stage design, or the first iteration of the product (Wania, 2008).

• Patterns helped designers to perform better than designers exposed to guidelines (Wania, 2008).

• Patterns may help in communicating interaction design knowledge (Saponas et al., Wania 2008)

Overall, all studies suggest that patterns have a positive influence, but we, as a community, might be looking for an impact in the wrong place, e.g., early stage design, design in
familiar domains. From the perspective of educating HCI practitioners, Griffiths and Pemberton’s (2000) suggestions sound appealing, where they call for teaching students on discovering and abstracting patterns. But, this may require more training and practice. What is also interesting, but not directly addressed in prior studies, is asking the question whether patterns help designers communicate interaction design advice, or whether patterns are a preferred format. The discussion in the next section converges on similar questions, and what might be leading, us as a community, to believe patterns might be a preferred format to capture and communicate interaction design advice or guidance.

2.9 Summarizing Empirical Work & Rationale for Proposed Research

This section summarizes the research on patterns in HCI, and introduces the scope and rationale behind the proposed research. It revisits the motivations behind the pattern approach in HCI, touching upon the need for a common language for design, looking for ways of capturing and communicating interaction design advice, enabling knowledge reuse, and to express design rationale. Arguably, the root of the challenge lies in capturing and sharing what we know about designing not only for members within the HCI community, but also across disciplinary boundaries.

The following discussion reviews why patterns may be argued for as a possible answer to these challenges. The proposed research explores the value of capturing interaction design advice as patterns to support communication. And in order to be effective, this advice needs to be communicated in a manner that could be understood and applied by its audience (Sutcliffe, 1996). Interaction patterns have been suggested as a format to document such interaction design knowledge.
2.9.1 Promises And Problems Revisited

Over the past few decades, patterns in HCI have received an increasing amount of attention. To its credit, the HCI community has over 400 published interaction patterns (Henninger and Correa, 2007) spanning broad areas of HCI (Tidwell, 2005), and also some specific areas like web-design (van Dunyne et al., 2003). The interest in patterns is not restricted to the research community in HCI, but has garnered interest from the practitioner community, e.g., Yahoo Pattern Library. For long the focus has been on the proposed benefits of using patterns in HCI, and discovery of patterns.

Review of related literature on patterns in HCI revealed four major themes, which represent the promises made: patterns as lingua franca, as a format to capture and share interaction design knowledge, enabling knowledge reuse, and as a way to express design rationale.
Patterns and pattern languages in HCI have been suggested as a possible way to compete with this growing complexity and diversity. Bayle et al (1997) present the case that HCI, as we know it, is changing and growing in new directions. Interactive systems are pervasive, and are now applicable in more contexts than before. They base their suggestions on the following properties of patterns:
• They are based on concrete prototypes drawn from the domain in which design is being done
• They work at multiple levels—community, group, individual—and they endeavor to tie the levels together
• They attempt to bridge the gap between the physical and social worlds
• They seem to be amenable to gradual, piecemeal development (p. 18)

Erickson (2000) believes that the concrete, socially-situated, and generative nature of patterns gives it the potential to serve as building blocks for a common language, or lingua franca, that could be used between designers and users to communicate, echoing Alexander et al. (1979) proposal for pattern languages in architecture. Borcher (2000) sees patterns as way to improve user participation in the design process.

Patterns have been proposed as a format for capturing and communicating interaction design advice (Griffiths and Pemberton, 2000; Borchers. HCI already makes use of design principles, and guidelines (Nielsen and Molich, 1994) to achieve this goal. Patterns are argued to be more specific than design guidelines, but more general than rigid style guides. van Welie (2000) asserts that guidelines are problematic in the hands of novice designers as they either tend to be too simplistic or abstract. In addition to this, there is no explicit design guidance to handle situations where guidelines conflict. The lack of specificity and context makes them hard to interpret. Borchers (2000) argues patterns are better than ‘golden rules of interface design’ because they contain concrete examples.

Patterns might allow designers to reuse interface design best practices, both for improving the quality of designs and also for educating novice designers. Grandlund et al. (2000) suggest HCI pattern resource could serve as an inventory of solutions to commonly encountered interface design problems. It is believed using patterns would allow inexperienced designers to

Patterns contain the essence of a solution to a recurring design problem in a specific context. Patterns also contain rationale that explains why it is a good solution, which is missing from other forms of interaction design advice (e.g. guidelines). All the different versions of interaction patterns, available in different pattern collection, include this rationale section. Moran and Carroll (1996) argue that design rationale explicate the assumptions, trade-offs, and negotiations, made about a design choice. They believe having access to this design rationale may prove valuable for communication between design stakeholders. This would enable designers to gain a deeper understanding of the consequences stemming from their design choices (Carroll, 2006).

As discussed earlier (refer figure 2-7.), there are some obstacles to be overcome before the promises could be realized, which include the following: lack of a commonly accepted pattern format in HCI, lack of an organizing principle for patterns, insufficient tool support, demonstrating empirical benefits of using patterns. More recent efforts have resulted in an XML based standard pattern format, PLML (Fincher et al., 2003), which holds promise for future tool support too. We are yet to see any empirical-benefits of various proposals for organizing patterns in HCI as languages or as collections. While research efforts towards a common format or tool support are encouraging, Dearden and Finlay (2006) believe it is important to demonstrate the empirical benefits of using patterns in the design process, or for the designed product. They feel empirical work a missing part of research on patterns in HCI, which warrants immediate attention.
2.9.2 Comments On Empirical Work

We have begun to see empirical work in the area of patterns in HCI, both in the form of controlled studies and case studies. The controlled studies appear to target the questions whether using patterns impact the quality of designed product (Chung et al., 2004; Saponas et al., 2006; Wania, 2008). The case studies have explored the use of patterns within a participatory design setting and also to teach HCI principles (Borchers, 2002; Griffiths & Pemberton, 2000; Kotze et al., 2008).

2.9.2.1 Empirical Work on Teaching HCI Principles With Patterns

The case studies attempting to teach HCI design principles using patterns summarize their findings as the following:

- HCI patterns lead to above-average retention of design principle (Borchers, 2002)
- Patterns are useful tools for teaching HCI design principles. (Borchers, 2002)
- HCI design principles should be taught by setting students practical exercises in using a pattern language in the process of design (Griffiths and Pemberton, 2000)
- The ability to notice successful solutions as patterns might be more useful skill than familiarity with set of patterns of a pattern collection (Griffiths and Pemberton)
- Students taught using patterns (positive guidelines) performed significantly better than those taught using anti-patterns (negative guidelines) (Kotze et al., 2008).

The reports on teaching HCI design principles tend to emphasize the positive aspects of using patterns. With the exception of the study by Kotze et al. (2008), the remaining few studies (Borchers 2002, Griffiths & Pemberton, 2000) base their findings on subjective measures.

Borcher’s used an in-class survey to assess usefulness of patterns. Students report patterns were very useful, but felt it did not help in their prototyping exercise. Borcher’s assertion about above
average retention of HCI design principles is debatable when no comparison group was present. Griffiths and Pemberton’s reflection about patterns are based on the feedback they received from students. No concrete measures were reported in their paper. Kotze et al. use final score (judged) awarded on assigned projects as an empirical measure to assess whether patterns or anti-patterns had an impact. But, their study was motivated by anti-patterns, and not patterns. More importantly, Kotze et al. operationalize the pattern concept as a guideline along with an example, arguing they were interested in comparing the effectiveness of negative interaction design advice. It is arguable whether interaction design advice missing a basic pattern structure, i.e., problem, solution, context and a diagram, could be called a pattern format. It raises doubts about whether their findings apply directly to teaching HCI with patterns.

2.9.2.2 Empirical Work on Using Pattern in Participatory Design Setting

The case studies on using patterns in a participatory design imply the following:

- Patterns enabled novice users to design, while supporting expert designers.
- Enabled the user-designer team to come up with coherent designs
- Participants found examples present in patterns helpful.

Dearden, Finlay, Allgar and McManus (2002a, 2000b) introduce patterns in a participatory design setting. This study used a facilitated session where the designs or prototypes were co-designed by the user and the facilitator. They report that users’ responses were positive, and once users became familiar with the patterns, they found it useful. But, they acknowledge that the positive responses could be attributed to any or a combination of all of three factors: use of a pattern language, paper prototyping exercise, and facilitator role. The authors were unable to comment on quality, as there was insufficient data.
2.9.2.3 Empirical Work Using Patterns and Impact on Quality of Product

The controlled studies suggest benefits of using patterns in design as the following:

- Participants rated usefulness of pre-patterns more for future projects than for the design activity (Chung et al., 2004).

- Participants rated usefulness of pre-patterns higher for design than for evaluation (Chung et al.)

- High experience designers report pre-patterns to be more useful for the design activity than low experience designers (Chung et al.)

- Pre-patterns would be useful to designers working with unfamiliar domains (Saponas et al., 2006)

- Patterns may not help early stage design, or the first iteration of the product (Wania, 2008).

- Patterns helped designers to perform better than designers exposed to guidelines (Wania, 2008).

- Patterns may help in communicating interaction design knowledge (Saponas et al., Wania 2008)

Chung et al. (2004), Saponas et al. (2006) base their findings on controlled experiments using pre-patterns. And although it may appear pre-patterns might have been helpful, their experimental findings do not seem to corroborate their statements. Chung et al. found no statistically significant differences in comparing the design task outcomes from the patterns and no patterns group. Saponas et al. found no significant differences in the level of completeness, detail, and overall quality of the prototype, between groups exposed to pre-patterns and those who
were not. Saponas et al. used professional designers as participants. They found that, for level of
detail, the control group performed better than the pre-pattern group. Level of detail was the only
dependent measure that was statically significant. Level of detail was an indication of the depth of
the prototype, or how readily implementable was the prototype. They, however, do not expand on
these findings, i.e., possible reasons why the control group did better on detail. The support
expressed for patterns in design appears to be derived from participant feedback, and subjective.

Another point of note is that both studies use pre-patterns. Pre-patterns, as stated by the
authors, are not patterns since they represent future interactions with systems in the ubicomp area,
i.e., these interactions are not available in current devices or systems. While patterns, by
definition, are backward facing, i.e., successful solutions to common problems in existing
systems, pre-patterns tend to be forward facing or future interactions. The implication is that
compared to patterns, where designers may have prior experience with the described
functionality, designers exposed to the pre-patterns might not be able to connect with the
experience or solutions suggested.

None of the above mentioned studies compare the benefits of using pre-patterns with any
other techniques used in HCI, e.g., guidelines. Wania (2008) shares the empirical measures for
assessing impact of patterns with the other two controlled studies. In addition to completeness,
detail, and overall quality, this study included ease-of-use as an additional dependent variable.
The author argues that since patterns are being discussed within the HCI domain, usability should
be considered as a desired attribute. Also, instead of using pre-patterns, this study used patterns in
information retrieval systems. These patterns were discovered and documented based on
interacting with real systems in use (Wania & Atwood, 2007). It also compared patterns with
guidelines (Nielsen and Molich, 1994). Wania found no significant differences in the expert
ratings for the paper prototypes designed by the patterns, guidelines or control group. However,
pattern group seemed to perform better than the guidelines group, and the control group. The control group here did better than the guidelines group, similar to Saponas et al. (2006) study. As discussed earlier, the Wania concludes that patterns may not help under the following conditions: first iteration of the prototype, and familiar task domain.

The author argues that first iteration, with individual participants, is a complex activity since the designers have to conceptualize the requirements and make several design choices. It is a first attempt at the design, and hence may not be an ideal prototype to evaluate for quality. The idea of reuse, promulgated within the HCI community, may have lead us to believe that having access to good design practices would allow designers to shave design iterations (and time) by leap-frogging over some sub-par design choices. This does not, however, imply that the gains would be apparent in the very first attempt at designing. Even if we may have access to good design practices, one still needs to consider a variety of design options before cementing these ideas in the design. In the light of this finding, Wania proposes that patterns might be helpful to make changes to an existing design, as now the designers can focus on improvements and not the initial design.

Familiarity with the domain also appears to have influenced how designers performed. This familiarity could also be interpreted as experience designing in a similar domain, or exposure to certain systems. In the case of Wania (2008), the task involved designing interfaces for an information retrieval system, and the patterns were derived from existing systems (e.g., google.com, portal.acm.org). The task was to design interfaces to support a regular search and an advanced search. Wania argues that although the participants may not have designed similar systems, their idea of what an IR interface may look like could be influenced by past experience using such systems. This might cause participants to rely less on a set of explicit patterns presented to them, but instead use pre-existing knowledge. The design community makes a
reference to this as design repertoire. The control group, in Wania’s (2008) study, appears to have performed better than the guidelines group, which may indicate prior experiences playing a mediating role. Whether we choose to call prior experiences as a pattern is debatable, but prior experiences also seem to be source, and part, of the pattern discovery process (Wania, 2008; Alexander, 1979).

These arguments seem to raise questions about whether patterns would be helpful in improving quality of design, especially the first iteration. The empirical work on patterns indicates that it may aid communication. Wania (2008) traces pattern-like structures in use in other domains, and make the argument that we may be already using it, i.e., problem, solution, context, without calling these patterns. This brings us back to one of the promises made about patterns, which hinges on the proposal that patterns may be a preferred format for capturing and communicating interaction design advice. The next section discusses why we may be lead to believe that a pattern structure might be helpful for delivering design advice.

2.10 Communicating Interaction Design Advice Using Patterns

As discussed in the previous section on empirical work on patterns in HCI, Chung et al. (2004), Saponas et al. (2006), and Wania (2008), appear to converge on patterns may be helpful for communication. This aligns with some of the promises (refer section 2.9.1) made about patterns in HCI. Pattern languages have been proposed as a common language for interaction design (Erickson, 2000; Borchers, 2000; Bayle et al., 1997). The pattern structure has been argued as a format to capture interaction design advice (Bayle et al., 1997, Griffiths and Pemeberton, 2000; Fincher, 2000; van Welie, 2000). It has been proposed that patterns may be potentially more helpful to designers since they include concrete examples that help situate the design advice (Borchers, 2000). Mahemoff and Johnston (1998) assert that patterns in HCI would allow for discussion of usability in context.
Promises made about patterns and lingua francas, patterns and knowledge reuse, and patterns and design rationale seem to hinge on pattern structure (i.e., problem, solution and context). It is generally assumed a pattern structure may be an effective way of communicating interaction design advice. Consequently, as shown earlier, this pattern structure has been adopted as the de-facto standard for all versions of patterns in HCI. When comparing with other forms of design advice, e.g., design principles or guidelines, the structured pattern content has been argued as its strength over other forms. Design guidelines are sometimes accompanied with examples (Dearden and Finlay, 2006), but lack the context piece (or a range of contexts), or the argumentation (why is this a good solution). The role played by pattern structure appears pivotal since this representation has been strongly suggested for capturing HCI design knowledge. And while discussions on patterns in HCI span a couple of decades, there is little empirical work demonstrating the benefits of capturing design knowledge as patterns. The community continues produce more patterns based on an assumption that a pattern structure might be useful for communication.

Saponas et al. (2004) observed how their participants made use of pre-patterns during the design activity for communication. Based on a protocol analysis, they determined patterns were used under four conditions mediating communication: idea generation (31%), re-reference (17%), discovery (17%), and issue clarification (35%). The numbers in the parenthesis indicates the percentage of instances coded under each category (out of 103 total instances). Idea generation and discovery seem to represent two ends of designer’s intent: looking for pre-patterns that could be inspire new ideas, or looking for patterns that fit the task, respectively. Issue clarification is defined as designers using pre-patterns seeking answers for an articulated problem. Re-reference dealt with going back to something the designers had encountered when perusing earlier. While the authors report their analysis, there is no connection made to the impact on design activity or
process. They do not report any analysis on the communication practices within the control group. Dearden et al. (2002) report patterns helped designers and users communicate design ideas, and the examples contained in patterns were helpful. Overall, the participants reported a positive experience from the design exercise, but there is no mention of empirical benefits of using patterns.

The following discussion will discuss the structure of the pattern advice adopted in HCI along with discussing how Alexander’s work on patterns and pattern languages has been interpreted. It will revisit the contrasts between patterns and other forms of interaction design advice. The question this discussion will explore is why might we, as a community, be lead to believe that the pattern structure is beneficial? Patterns have also been suggested as a way to capture design rationale (Dearden & Finlay, 2006). Structured formats for expressing design rationale have already been discussed in the past (Moran and Carroll, 1996). What makes patterns different from other design rationale notations for expressing rationale? What role does a pattern structure play in communicating design experience?

2.10.1 Pattern Structure in HCI

Alexander (1977, 1979) had proposed pattern languages to be used by architects and non-architects to communicate. Patterns were suggested as building blocks for such a language. Alexander describes the essence of a pattern as, “Each pattern is a three-part rule, which expresses a relation between a certain context, a problem, and a solution” (p. 247). This structure is present in pattern descriptions in HCI, and also other disciplines (e.g., software engineering). Alexander’s contribution could be seen as suggesting a way to discover and document patterns, and show how patterns could exist as a language. “A Pattern Language” could be argued as Alexander’s approach for showing that patterns exist, and there are relationships are between them. By specifying these relationships, one could potentially construct a language for design.
The intent was to make design knowledge both shareable, and understood by a wider audience, not only architects.

Table 2-3 Showing the basic pattern structure of problem, solution, and context present in different versions of patterns in HCI

<table>
<thead>
<tr>
<th>Source</th>
<th>Problem</th>
<th>Solution</th>
<th>Context</th>
<th>Rationale</th>
<th>Image/Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tidwell</td>
<td>What</td>
<td>How</td>
<td>Use When</td>
<td>Why</td>
<td>Examples</td>
</tr>
<tr>
<td>Van Welie</td>
<td>Problem</td>
<td>Solution, How</td>
<td>Use When</td>
<td>Why</td>
<td>Examples</td>
</tr>
<tr>
<td>Van Duyne et al.</td>
<td>Problem</td>
<td>Solution</td>
<td>Background</td>
<td>Problem description</td>
<td>Examples</td>
</tr>
<tr>
<td>Yahoo</td>
<td>Problem</td>
<td>Solution</td>
<td>Use When</td>
<td>Rationale</td>
<td>Examples</td>
</tr>
</tbody>
</table>

Alexander’s proposed approach may have prompted criticism from within the architectural community (cf. section 2.5.1), but there seems little, albeit non-zero, resistance within HCI. It could be argued that HCI does not share artifacts with architecture, and hence may not find design advice presented as particularly useful. But, architects may differ on the claims made through patterns. Some within the HCI community have raised questions about whether it is possible to capture a timeless way of designing for a fast-changing field like HCI (Borchers & Thomas, 2001; Griffiths and Pemberton, 2001), or whether it is reasonable to adopt scale as an organizing principle (Borchers, 1999; van Welie & van der Veer, 2003). Using scale, in the geometrical sense, may prove limiting in HCI where screen-real estate is only a part of design discussion. Other variables, e.g., temporality, sequence of tasks over time, also needs to be considered.

Influence of Alexander’s proposal on HCI could also be interpreted as a way to approach design and to address design complexity (Bayle et al., 1997). Alexander (1979) argues that “Everybody follows rules of thumb” (p. 201) during design, which he refers to as patterns that are
accumulated over time and through experience. When faced with a problem, we make use of these guiding principles to support our decisions since it would be difficult to start from scratch every time (Alexander, 1979). These “patterns” help designers cope with, and be efficient, when dealing with the design complexity. Such arguments lie at the heart of Alexander’s proposed design approach. Patterns were offered as a structure to capture this knowledge so that it is shareable. In a larger context of design methodology across disciplines, Alexander’s works are viewed as related to design complexity, and design rationale (Atwood, McCain, and Williams, 2002). At a more restricted scope, within HCI, his works are also viewed as contributions to design theory and as way to approach design complexity (Wania, Atwood, and McCain, 2006), placing him closer to others like Donald Schön and Chris Agyris. Figure 2-8 shows the results of an author co-citation analysis of cited authors in HCI. ACA belongs to a larger body of techniques devoted to the quantitative study of academic literature or Bibliometrics (White & McCain, 1998). ACA attempts to show hidden relationships between authors based on the number of times two authors are cited together in one document (in the reference section). As explained by Atwood, McCain and Williams, these relationships attempt to show how the community as a whole views authors’ oeuvres, and not individual pieces of work.
2.10.2 Other Formats Used for Communicating Design Advice

HCI principles (Norman, 1986), guidelines (Nielsen & Molich, 1994) and style guides (Apple, GNOME) have been used to communicate HCI design knowledge. Design guidelines have been called abstract since they are too general (Griffiths and Pemberton, 2000). They also lack context, i.e., when to use a particular guideline (van Welie, 2000). van Welie (2000) summarizes the problems faced while using guidelines as being too simplistic, difficult to select and interpret, and often conflicting when multiple guidelines may be applicable.
For example, consider a task of creating a login window, which is now common on websites to authenticate access to restricted content. The following guideline (Nielsen, 1994) could be useful in this case:

**Recognition rather than recall:** Minimize the user’s memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

![.Mac login](image)

Figure 2–9 An example log-in screen, which reminds the visitor the kind of information expected.

Figure 2-9 shows how the guideline “Recognition rather than recall” could be used to evaluate this interface. Above the text input field, visitors are reminded to enter their login information in a particular format. The guideline by itself, however, provides no hint to the reader that it could be instantiated in this manner or would be useful in this context. Reflecting more may show other guidelines applicable here, e.g., error prevention. The design guidelines also appear to miss rationale, which may be required to foster a deeper understanding or learning, especially when we may not always be dealing with expert designers.
In style guides, design advice is hard-coded to a particular context or a specific environment, e.g. Apple user interface guidelines are meant for the Apple OS X user interface. Style guides tend to be very detailed, and contain prescriptive design advice (Mahemoff and Johnston, 1998). There appears to be little room for deliberation or discussion, since there is one prescribed way of designing a feature, which helps to maintain consistency in the user interface. Style guides need to be specially drafted, and hence are not available for all types of interactive system. But, in both cases, the onus and success of refactoring style guides for another context, or reasons for selecting relevant guidelines, seems to depend on the skill and experience of the designer. The specificity of design advice in style guides, or the generality of advice present in guidelines, may demand more of the designer (Mahemoff and Johnston, 1998).

2.10.3 Structuring Design Advice With Rationale

Providing rationale along with design guidance is not unique to patterns. Design rationale research has proposed several structured representations in the past (Moran and Carroll, 1996) to capture and represent rationale. These representations attempt to serve multiple purposes: communication, coordination, and even as an explicit record of design knowledge about both process and product (Carroll & Moran, 1991; Moran & Carroll, 1996; Regli, Hu, Atwood, & Sun, 2000).

It’s easy to see why it might be difficult to produce a one-size-fits-all explanation, which addresses all forms of post-hoc inquiry about the design. It would be impossible to have perfect foresight to determine what, how and when explanations would be demanded. It may also depend on who is producing this rationale, and for whom. By nature, the explanation produced may be of greater use to people other than the person who authored the explanation (Grudin, 1994), which may be the intent of producing the rationale in the first place. Even if we claim to have such foresight, as Horner & Atwood (2006) point out, there can be several reasons behind producing
incomplete rationale, intentional or unintentional. An empirical study on usefulness of rationale
documents produced show that such documentation helped answered only half of the question
raised by designers about design decisions made (Karsenty, 1996). While this not damning
evidence against rationale recorded, it highlights some of challenges mentioned earlier.

There are some properties of patterns that appear to differentiate it from other forms of
design rationale: capture best practices; reflective rationale; levels of abstraction; treatment of
context, design guidance; and availability.

• Unlike other forms of design rationale (e.g., QOC (Maclean, Young, Bellotti &
  Moran (1996) or PHI (McCall (1991)), patterns attempt to capture what works or
the best practices. It has less to do with capturing history of the design
deliberation (e.g., PHI), structuring deliberation (e.g., IBIS) or for documenting
the design options considered (e.g., QOC). DR notations could be used to explain
the justifications behind a good or bad decision. The rationale presented in
patterns provides justifications for the problem and solution, and usually for a
good solution (anti-patterns document the opposite).

• Documenting design rationale is hard, especially during the design activity
  (Fisher et al., 1991). But, unlike the other DR structures, patterns represent a
post-hoc or reflective form of design rationale (A parallel for a reflective form of
DR could be found in claims analysis), which is discovered in successful
implementations. Discovering and scoping a pattern is no easy task, but capturing
rationale for something that is tangible and exists in systems may be easier than
capturing rationale for functionality that is being deliberated.

• Patterns are available at different levels of abstraction, and the rationale is
  available within each pattern description (e.g., E-commerce site, homepage,
breadcrumbs, tab rows). PHI or QOC appear to be more project specific instead of dealing with specific interactions.

- Patterns explicitly attempt to encapsulate a range of contexts within which the problem-solution pair exists. DR notations implicitly capture context of the design decision by situating within the larger discussion or project. Deliberations tend to be focused on the task at hand, and not driven by the goal of repurposing or generalizing the design advice in other contexts.

- Patterns attempt to present design guidance, or explain how solve the design problem. The rationale (i.e., argumentation) in this case is part of the pattern description, and not the only intent. DR is primarily concerned with capturing rationale about the design activity.

- And finally, patterns appear to have more traction within the HCI community compared to other DR notations. With more than 400 patterns in HCI spread across 12 collections, there is incentive in trying to understand the empirical benefits behind using a pattern structure.

2.10.4 Pattern-Like Structures Already in Use

Earlier it was argued that there is not much evidence in HCI showing a pattern-structure is effective for communicating design advice. Wania (2008) suggests that pattern-like structures have been used in design communication, but have never been referred to as patterns. Among the instances she provides, the following are particularly interesting: patterns as reflective discussion, war stories used by Xerox repairmen, and Explanation patterns.

Wania revisits a discussion between a master and apprentice, talking about a design problem, in *Educating the Reflective Practitioner* (Schön, 1987, cf. p.44). Wania argues that this
conversation, and the explanation provided by the master, spans discussion of the design problem, the context in which it exists, the resulting drawings and a possible solution. She asserts there are “… obvious similarities between patterns and the conversation that is taking place in the design studio, but, also obvious similarities between patterns and what Schön is describing as being contained in the designer’s repertoire” (p.130). She implies that the pattern structure may in fact be a natural structure present in design communication.

Orr’s (1996) observations, on how XEROX repair technicians exchanged their experience troubleshooting copier problems, highlighted the role played by war stories. Orr describes war stories as “anecdotes of experience” (p. 125), which basically described how a technician solved a problem related to a particular kind of machine. These stories served as the format that was circulated around, by which technicians shared their knowledge and experience. Wania (2008) points out the presence of a problem (copier trouble), context (type of copier), and the solution (how it was fixed) in these stories. The EUREKA system, a knowledge management system, later implemented at XEROX made use of this grass-root level knowledge, war-stories, for sharing within the community of practice (Bobrow ad Whalen, 2002). EUREKA system is often referred to as a successful knowledge management implementation (Malhotra, 2002). Wania argues that there might be a relationship between capturing experiences in this format, and the success of the EUREKA system, which made use of this structure too.

When talking about nature or structure of explanations, Schank (1986) explains that humans rely on previously used explanations to construct new ones, in the same way that we rely on past experiences to make sense of new ones. He argues there is no guarantee that these past experiences would lead us in the right direction, or useful interpretations, but that does not prevent us from making these leaps in judgment. Schank argues for explanation patterns (XP) that help us construct these new explanations. He describes it as,
... a fossilized explanation….When it is activated, it connect 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)

He goes on to describe an explanation pattern (XP) in greater detail (c.f. Schank, 1986, pp. 111-112). For the sake of brevity, the following bullet point highlights some of the key aspects raised by Schank, which indicate a similarity with pattern-structure:

- Index to the pattern, consisting of a combination of states and events. This is followed by a set of states of world when these indices are active.

- A scenario, which is a story that lays out the chain of events, or a plan of action for activating the combination present in the index.

- The resultant state following the scenario, along with other explanations generated using this XP.

There appear some clear parallels between using past experiences in design for new design, and using past experiences to interpret new ones. The bullet points shown above seem to relate to context that would trigger our memory, the scenario relates to a problem description, and the explanations generated in the past as a possible solution. Wania believes there are even similarities between how designers would use past experiences to bear upon existing design problem, and also how interaction patterns might be retrieved and used in design.

### 2.10.5 Summary of Patterns and Communication

Overall, the discussion presented touched upon the following topics:

- Empirical work on determining the impact of patterns on quality of the designed product is inconclusive. But, all agree on the benefits of patterns for communication between designers, and for communicating design advice.
• Alexander’s pattern structure has been adopted by and large by the HCI community for documenting patterns for HCI

• Much has been said about the benefits of the pattern structure over other forms of design advice. To a large extent, the presence of context of the design advice, and rationale for why it is a good solution, have been argued as factors dictating the HCI community’s interest.

• Documenting rationale is not unique to patterns, and structured formats are not a novel concept. What is new here? Patterns, however, are proposed not only for capturing design rationale, but also for providing design guidance.

• Patterns seem to be popular, and have more traction within the HCI community, compared to these other structured formats. This is an incentive for pursuing the proposed work.

• Pattern-like structures can be traced in other disciplines, as a natural structure for design communication, and communicating experiential knowledge. This indicates there might be some merits in using a pattern structure.

This section presented arguments that patterns structure may be useful for documenting and communicating interaction design advice. But, we have not seen any empirical work in this area, which would suggest patterns are an effective format for capturing interaction design advice. This motivates the research questions explored in this proposed research.

2.10.5.1 Research Questions

The broad question this research explores is whether a pattern-like structure (problem, solution, and context) is an effective format for explaining and communicating interaction design advice, that is, do interaction patterns help interaction designers communicate and share
interaction design knowledge? Furthermore, how does the structure of advice affect communicating design advice?

This present study attempts to evaluate the impact of a pattern-like structure on delivering usable and useful design recommendation, addressing design trade-offs, and capturing the context where this advice is applicable. These aspects are presented below as questions motivating this research:

1. How does the advice structure affect the effectiveness of design advice?
   a. To what extent does the advice structure impact the context of use described?
   b. To what extent does the advice structure impact the rationale expressed?
   c. To what extent does the advice structure impact the usefulness of advice?
   d. To what extent does the advice structure impact the overall quality?

2. Does the advice structure affect the time required to prepare design advice?

As argued earlier, similar to design rationale, documenting patterns may be meant for readers who are not necessarily authors. Molich, Jeffries, & Dumas (2007) claim that even popular interaction design textbooks in interaction design fall short on explaining how to effectively communicate design recommendation. They believe the emphasis is on training novice designers to understand and apply a user-centered design process (Gould & Lewis, 1985), or training designers on empirical and inspection methods for finding usability problems. And while designers may succeed in finding majority of usability problems using these techniques, Hornbæk & Frøkjær (2005) suggest that developers appreciate re-design proposals more than a list of usability problems, even when problems have assigned severity. Wixon (2003) asserts that problem detection alone does not guarantee improvement in product quality. Wixon's call for case-studies could be interpreted as a need for explaining the process of problem identification, i.e., an explanation of recommendations and interventions that worked, situated in a specific
context (organizational). Jeffries (1994) and Dumas, Molich, & Jeffries (2004) argue for presenting concrete problem descriptions along with concrete solutions. Jeffries suggests that recommendations should include justifications for both the problem and solution. Since no solution is a perfect solution, the recommendation should also address trade-offs that are inevitable.

Molich et al. (2007), Molich, Pero, Modgil, & Schroeder (2006); Molich, Hornbaek, and Scott (2007) suggest that quality of the design recommendation depends on the concreteness and detail of the suggested solution, as much as being an effective solution (address usability). They refer to this as the usefulness and usability of design recommendation. Expectations from design recommendations can be summarized as the following:

- Usable, i.e., have necessary detail present so that readers have concrete suggestions to work on.
- Useful, i.e., communicate an effective idea that addresses a usability problem
- Include justifications or rationale behind the design advice, i.e., the limitations and trade-offs of implementing the recommendation

2.10.5.2 Comparing Pattern-Advice to Claims-Advice in the Study

Claims were selected as one of the comparison structures because of the similarities between claims and interaction patterns. Both, patterns and claims have been proposed in the literature for describing rationale behind the design decisions. Similar to patterns, the process by which claims are produced tend to be more reflective in nature when compared to other proposed design rationale (DR) notations (e.g., QOC, IBIS). The other design rationale notations seem to be better suit describing the process of design (argumentation and deliberation) or structure of the design space. Moreover, recent literature include proposals for using claims for capture of HCI
design knowledge (Sutcliffe & Carroll, 1999; Sutcliffe, 2000), and as a usable format for communicating design advice (Fabian et al., 2004; Fabian, Wahid, Bhatia, & McCrickard, 2006; Sutcliffe & Carroll, 1999; Sutcliffe, 2000). This appears to mirror suggestions and promises made about interaction patterns. Claims have been shown to be a useful HCI technique in various studies, both as a part of the scenario-based design (Rosson & Carroll, 2003) and also by itself (McCrickard & Chewar, 2003; Venturi & Bessis, 2006).

Table 2-4 compares the different formats available in the literature to capture design rationale. These formats have been compared along six dimensions described below:

- **Design Space**: Mapping solution alternatives for the given design problem. In other words, documenting the different possible ways a problem can be solved. These solutions can be empirical (i.e., derived from existing interactions) or analytical and novel (i.e. new solutions that did not exist before).

- **History**: Tracing how a final design decision was arrived upon. That is, maintaining a history of not only the solution alternative selected from the design space, but keeping a record of solutions that were not selected and its evolution based on iterations.

- **Trade-offs**: Identifying participating design issues and visualizing the inter-relationships.

- **Reflective**: Constructing design advice or guidance as a synthesis of the design deliberations, after trade-offs have been made decided on. This is done for the design alternative that was selected amongst the alternatives. The synthesis represents the unit of information or artifact, which may be re-used as design communication instead of the design space or history itself.
• **Abstraction**: Capturing design guidance at different level or scope of interaction. The guidance can be about a system wide decision or more holistic (e.g., deciding on an interaction framework) or be about a very specific interaction (e.g., best practices in displaying form labels on forms)

• **Guidance/Advice**: Related to “reflective” dimension, but here it means whether the guidance is targeted at solving a specific design problem, which is usable (i.e., concrete and actionable) and useful (i.e., relevant design issues and implications are highlighted) recommendations

Using this framework, patterns and claims seem to share most of the dimensions when compared to other design rationale notations. In particular, both patterns and claim-advice highlight trade-offs, are constructed post-hoc as a synthesis of the design decision, can exist at different levels of abstraction and contain actionable design advice.
Table 2-4 Highlighting the similarities/differences between patterns, claims, QOC and PHI techniques with respect to intent and strengths of each approach.

<table>
<thead>
<tr>
<th>Structure</th>
<th>Design Space</th>
<th>History</th>
<th>Trade-Offs</th>
<th>Reflective</th>
<th>Abstraction</th>
<th>Guidance/Advice</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATTERN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QOC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PHI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLAIMS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

While there are similarities between claims and patterns, there are also differences. Claims focus on the trade-offs, i.e., pros and cons of implementing the design feature. The context is implicit in the scenario describing how the design feature would be used. Patterns aim to draw attention to an interaction design problem, the essence of a solution that addresses this problem, and emphasizes the context in which the problem-solution pair exists. Patterns are problem-driven, while claims are not. In a pattern structure the context tends to be explicit. But the trade-offs, or constraints, imposed by the context on the solution appear to be implicit. Patterns tend to discuss the relationship between the problem, solution, and context instead of focusing on context in abstract.
Dearden and Finlay (2006) argue that patterns have not been compared to any other HCI technique, and comparing to a control group alone may not allow us to learn much. Wania (2008) compared patterns with guidelines for a design activity. Guidelines were not considered in the present study because it does not include context or rationale. More importantly, participants would be asked to produce design advice, and not guidelines. Dearden and Finlay (2006) had compared patterns to claims based to the similarity.

The following chapter discussed the research method proposed for answering the questions presented in this section.
CHAPTER 3: RESEARCH METHOD

3.1 Introduction

As discussed in earlier sections, there has been a lot of interest surrounding patterns for HCI. It has been argued that patterns have the potential to serve as a common language for interaction design, allow reuse of HCI best practices, and provide a structured format to capture these best practices, which addresses both the context of the design advice and rationale behind it. Lack of tool support, disagreement over a common format, and differences over an organizing principle are some of the challenges observed. But, empirical work showing benefits of patterns to the designed product or the process has been known to be lacking. The few related empirical studies that were discussed earlier are inconclusive about use of patterns and impact on product quality. However, it is still suggested that patterns may be valuable for communication, and for capturing and communicating interaction design advice. Based on this premise, the pattern structure is used as the de-facto format for documenting interaction design advice. But, we have little empirical evidence demonstrating the benefits of using this structure. The pro arguments suggest that inclusion of context and rationale along with design guidance would render patterns superior to other forms of design advice (e.g., HCI principles, guidelines). The purpose and scope of the proposed work involves an investigation of these assertions about a pattern structure. This section discusses the research method that would used to empirically evaluate the effectiveness of a pattern structure. It describes the research method proposed, which details the following:

- Participants
- Research Design
  - Independent variable
  - Dependent variables
- Instruments used to document the study outcomes
3.2 Research Questions

The broad research question examined in this study is whether patterns are effective in capturing and communicating design guidance. More specifically, the study investigates the impact of pattern-like structure on the usefulness and usability of design advice produced. This structure was also compared to a competing claims-advice structure, which shared similarities with patterns. The broad questions were operationalized by the following research questions:

1. How does the advice structure affect the effectiveness of design advice?
   a. To what extent does the advice structure impact the context of use described?
   b. To what extent does the advice structure impact the rationale expressed?
   c. To what extent does the advice structure impact the usefulness of advice?
   d. To what extent does the advice structure impact the overall quality?

2. Does the advice structure affect the time required to prepare design advice?
3.3 Participants

The participants for the study were recruited from the College of Information Science and Technology (IST) at Drexel University. The participant population included undergraduate and graduate students who had some HCI experience. That is, for inclusion in the study, all participants should have had completed at least one introductory course in HCI. The introductory course covers HCI design principle for user interface layout, principles of user-centered design, and at least one method for evaluating usability. Participants were recruited via information posted on Drexel University bulletin boards using flyer, and from classroom. It was made clear that participation is voluntary. Participants will be compensated monetarily for their time. Participants were recruited from College of IST students for two obvious reasons: ease of access, and similar HCI training (similar course content). Prior design experience was assessed using a pre-task questionnaire (see Appendix A).

3.4 Research Design

The intent of the proposed study is to study the effectiveness of structure of advice on communicating interaction design advice. A between-subjects experimental design was used to achieve this goal. Participants were randomly assigned to one of three experiment conditions: patterns group, claims group, and control group (shown in table 3-1).
Table 3-1 Table showing the three levels of the independent variable, structure of design advice, and related treatment

<table>
<thead>
<tr>
<th>Condition</th>
<th>Assignment</th>
<th>Treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern group</td>
<td>Random</td>
<td>Provided a pattern-like structure for completing design task</td>
</tr>
<tr>
<td>Claims group</td>
<td>Random</td>
<td>Provided a claim structure for completing design task</td>
</tr>
<tr>
<td>Control group</td>
<td>Random</td>
<td>No explicit structure provided, but receive identical design task.</td>
</tr>
</tbody>
</table>

In the remainder of the discussion using the word “patterns” will imply “pattern-like” structure. A detailed description of the independent variable under study follows.

3.4.1 Independent Variable
The study manipulated a single independent variable, structure of design advice, with three levels: pattern, claims, and control. A claims format, in addition to the control group condition, was selected for reasons discussed earlier.

3.4.2 Dependent Variables
This study evaluates the empirical benefits of pattern/claim structure on the interaction design advice produced. As part of the design task, participants were asked to make a design recommendation, i.e. a form of design advice. The design task is discussed in more detail under materials and instruments. Design advice prepared by the participants was evaluated on the following measures to assess the impact of structure:

- Context of use described in the advice
- Rationale Expressed
• Usefulness

• Overall Quality

• Time required to prepare advice

Context of Use Described (RQ 1.1) was a measure used to capture the level of concreteness and detail present in the design advice. CUE studies or Comparative Usability Evaluation studies use professional usability teams to independently evaluate a website or application. One of the objectives for conducting these studies is to explore the differences in the quality of usability problems and recommendations generated by each team (Molich et al., 1999; Molich & Jeffries, 2004; Molich, Jeffries, and Dumas, 2007). The first of the two measures used in this study are borrowed or modeled on Molich, Jeffries, and Dumas (2007) CUE-4 study.

Context of Use, as used here, is similar to the “Usability of Design Recommendation” suggested in the CUE-4 Study. Molich et al. argue for concreteness of the advice as a desirable attribute. Lacking this property may reduce the chances a change would be implemented. This recommendation also includes avoiding usability jargon (e.g., affordance, visibility) when describing recommendations. They defined a fully usable recommendation as,

Fully usable: The recommendation communicates precisely and in reasonable detail what the product team should do. In other words: About as good as it’s going to get, given the request for a short description. (p. 166)

They define an unusable recommendation as,

Usable: The recommendation is totally vague, unclear, or incomprehensible for the product team. The recommendation leaves all decisions regarding the implementation of the solution to the product team. This could introduce new usability problems in the solution. (p. 167)

Usefulness of the design advice (RQ 1.2) is defined as an effective idea that solves a usability problem (Molich, Jeffries, & Dumas, 2007). Any recommendations should take into
consideration other contextual factors that may interfere with implementing this recommendation.

A useful design advice should look beyond achieving a local optimum, and consider the design recommendation as part of a whole (application) (Molich, Jeffries, & Dumas). It is possible to have a usable, or concrete, recommendation that is not useful because implementing it may lead to more problems. They define a fully useful recommendation as.

\[
\text{Fully useful: Describes an effective idea for solving the problem. Contains no bad elements. In other words: About as good as it’s going to get, given that the instructions requested a “short description.” (p. 165)}
\]

On the other end of the spectrum, a not useful recommendation is defined as,

\[
\text{Not useful or misleading: Describes an idea that would not increase or might even decrease the usefulness of the product. This classification is also used when the recommendation is so vague or unclear that we do not understand it. (p. 166)}
\]

\[
\text{Rationale (RQ 1.3), as the name suggests, implies that the recommendation should include why is it a good solution for the given problem and context. Rationale provides the justifications that would persuade the readers, and help them to understand the reasoning and assumptions made by the designers in making their choice (Moran and Carroll, 1996). This variable would measure extent to which constraints or limitations of the solution and context have been addressed. Since no solution is a perfect solution, the recommendation or advice should also address the trade-offs of the solution (Jeffries, 1994). The rationale can be motivated by usability concerns.}
\]

\[
\text{Overall Quality (RQ 1.4) will be used to judge how good was the recommendation on the whole, and is decided based on the judge’s expertise on the subject.}
\]
3.4.3 Materials & Instruments

This section describes the instruments that were used to administer the task and collect data. Figure 3-1 provides a quick overview of the study procedure that helps situate the study materials discussed here. Based on their assignment (i.e., pattern, claims or control groups) participants followed an almost identical workflow. This involved completing a pre-task questionnaire, followed by a practice task to give participants some familiarity with the study expectations. Then they completed the main design task, outputs of which were captured in the data-collection sheet. The participant activity was considered complete after finishing the post-task questionnaire.
3.4.3.1 Pre-task Questionnaire

This questionnaire will be used to gather some demographic information about the participants and their background:

- Age
- Gender
- Degree pursued
- Current major
- Number of HCI courses completed at IST
- Number of HCI courses completed outside IST
- Experience designing systems (time in months)
3.4.3.2 Practice Task

A practice task was provided to participants as a pre-cursor to the main design task. The goal was to familiarize them with the structure they used in the main design task. Here, participants were asked to provide their design recommendations for an icon to be used on a mobile phone. The task provided some basic information about the context of use, the structure participants needed to use when offering their recommendation, and three solution alternatives to pick from, to mimic the actual design task (i.e., menu recommendations for room-planner). All participants spent a fixed amount of time for the practice task.

![Practice task](image)

Figure 3–2 Practice task asking participants to recommend an icon for the mobile phone task context.

3.4.3.3 Design Task

To understand the impact of the structure of advice on the effectiveness of the advice, the present study simulated a situation where the participant had to give a design recommendation for a room planner interface. Design recommendations have a lot in common with design advice since it not only involves suggesting a particular design choice, but also includes an explanation for why a particular design option is preferred.
For the experiment, the participants were asked to recommend a *menu* design for displaying a product catalog on a room planner interface. All participants received task instructions shown in figure 3-3 below. A room planner task context was chosen because the participants may be familiar with the process of furniture shopping, or what this system would help users achieve. This helped reduce how much participants needed to know about the task domain, and focus them quickly on the task,

A room planner allows shoppers to add products from the catalog to the room plan and try out different arrangements. It would help shoppers avoid the hassle of ordering furniture that is too big or too small for their homes.

You have to choose one among three menu layouts for presenting a product catalog on the room planner interface. You will be provided with the options. Explain the reasoning behind your choice using the provided instructions…

Figure 3–3 Part of the instructions provided to participants about the context of the design task.

The instructions showed a graphic room planner interface devoid of any interface controls to interact with it. This interface is shown in figure 3-4.
Participants were provided with three design options instead of asking them to come up with their own. Providing participants with the design options offered two potential benefits: participants can focus on explaining their recommendation without the distractions of designing a menu; limiting the possible number options participants can discuss may help control the variability in the recommendations produced, which may otherwise affect the evaluation phase (described later).

The decision for using three design options instead of using one, two, or ten was made in the interest of time and cognitive effort. Tohidi et al. (2006) argues that having at least three options allowed participants to be more critical, which is what was desired in this study. For the
experiment it was preferred that the participants spent more time working on their explanation behind their choice rather than spend time on picking an option from a large variety. For analysis purposes, the menu choice picked was less important than the recommendation they produced.

Figure 3-5 shows menu options made available to the participants. All menus were hand-drawn to contain similar levels of precision and information. Screenshots of real menus might not allow for this level of control. The menus selected were not novel by any means, and are present in interactive systems encountered by students.

Figure 3–5 Three menu options are available to the participants: (a) Directory Tree Menu, (b) Drop-Down Menu, (c) Accordion Menu

To compensate for lack of precision, and interactivity, each menu option included descriptive information about the menu (shown in table 3-2). These were included so that participants may understand the menu functionality enough to critique these options. However, care was taken not to provide excessive information about any one particular menu. Also, the
menu options were not radically different from each other, which may prompt participants to be more reflective when making their choice instead of making a design choice obvious. The menu option sheet provided to the participants is shown in Appendix E.

Table 3-2 Descriptive information provided alongside the menu options

<table>
<thead>
<tr>
<th>(A) Directory-Tree</th>
<th>(B) Drop-Down</th>
<th>(C) Accordion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expand and collapse different sections using +/-</td>
<td>Drop down reveals the product categories (down arrow)</td>
<td>Clicking on the arrows expands/closes the section.</td>
</tr>
<tr>
<td>Multiple sections can be opened at a time</td>
<td>Only one section open at a time</td>
<td>Only one section open at one time.</td>
</tr>
<tr>
<td>Product information and image displayed inside the menu</td>
<td>Product information and image displayed inside the menu</td>
<td>Product information and image displayed inside the menu</td>
</tr>
<tr>
<td>Fixed height for the menu</td>
<td>Fixed height for the menu</td>
<td>Fixed height for the menu</td>
</tr>
<tr>
<td>Scroll (outside) to view the entire open catalog</td>
<td>Scroll (inside) to view more products</td>
<td>Scroll (inside) to view more products</td>
</tr>
</tbody>
</table>

3.4.3.4 Structure of Design Advice

Once the participants made their choice, they were asked to provide (write down) their explanation using a provided structure, depending on the condition assigned.

In addition to the task context shown in figure 3-4 earlier, the pattern group received addition instructions shown below.
Address the following questions when explaining your decision:

a. What is the user trying to do?

b. How will the recommended menu option be used?

c. Explain why it is the best choice in this given situation?

Explain your decision as if you are teaching some one else to make a similar choice. Others reviewing your recommendation may not see the options that you rejected. Mark (a), (b) and (c) in the body of your explanation.

Questions a, b, and c map into the problem, solution and context sections of the pattern-structure respectively. The questions are phrased like so for improving comprehension. In essence, these questions attempt to ask what is the problem, what is the solution and in what context.

Participants in the claims condition receive the following instructions:

Address the following questions when explaining your decision:

a. How will the recommended menu be used?

b. Advantages of using your design choice

c. Disadvantages of using your design choice

Explain your decision as if you are teaching some one else to make a similar choice. Others reviewing your recommendation may not see the options that you rejected. Mark (a), (b) and (c) in the body of your explanation.

The claim structure was derived from different definitions offered for claims (Sutcliffe & Carroll, 1999; Sutcliffe, 2000), and from claims as it has been used in studies. The present study uses the basic structure of a claim, not the revised format that includes 10+ sections for using claims to describe a design feature (Sutcliffe, 2000). A similar approach has been adopted in
preparing the pattern structure for this study, where the basic structure of problem, solution and context was used.

The **control group** will receive the following set of instructions:

Explain your decision as if you are teaching someone else to make a similar choice. Others reviewing your recommendation may not see the options that you rejected.

Each participant received a worksheet to write or type out his or her recommendation. The worksheet contained instructions, depending on the condition, a section to rank order the three menu options shown in table 3-2, and a fixed rectangle area to provide their recommendation. Copies of the instruction sheet, menu sheet, and the worksheet are available in Appendix B, E, and G respectively.

3.4.3.5 *Post-task Questionnaire*

This questionnaire gathered participant feedback about the task completed (as shown in Appendix X).

- Level of agreement or disagreement about the following questions (Strongly Disagree-Strongly Agree)
  1. Understanding of the structure provided (Claims or Pattern condition)
  2. Whether the structure helped in explaining their recommendation
  3. Usefulness of the structure for the participant in the future
  4. Understanding of the instructions
  5. Whether participant enjoyed the activity

- Open ended questions
What the participant liked about the activity?

What the participant disliked about the activity?

The control group also received a post-task questionnaire similar to this one, but did not include questions 1-4.

3.4.3.6 Evaluation Worksheet

This worksheet was used to collect ratings data completed by the expert evaluators. It consisted of two parts. One was a reference sheet for the expert evaluators to consult while the completed their evaluations (c.f. Appendix L). The reference sheet defined the dependent variables used in this study. Along with this, for each dependent variable, two excerpts were shown, one showing a good instance and one poor instance for the variables. The reference sheet was structured as follows:

- Metrics Used and high-level definitions:
  - Context of Use
  - Usefulness
  - Rationale
  - Overall Quality

- Metrics explained with examples
  - Context of Use with a detailed explanation
    - Good Example
    - Poor Example
  - Usefulness with detailed…

The reference sheet showed an example that was previously rated by the researcher based on how the metrics were explained earlier so as to provide a concrete example. This example recommendation was presented in a format that was identical to how the expert evaluators
received the data resulting from the study (i.e., design recommendations). Appendix L and M show the reference document and how data was presented to the evaluators, respectively. The evaluation work sheet was integrated with each recommendation for ease of rating.

3.4.4 Procedure for Design Activity

Before beginning the experiment, each participant answered a number of questions related to their educational background and work experience. After this, participants were introduced to the practice task, for which a maximum of ten minutes was allotted. The practice task prepared the participants by introducing them to the structure used for preparing their recommendations. After completing the practice task, irrespective of the condition, each participant received the identical design task where they will be asked to recommend a “menu” design. Within the task context, this menu will be used to show a furniture catalog on a room-planning interface. A room planer was defined as an interface that will allow online shoppers to try out different furniture arrangements on a floor plan, before buying.
Figure 3–6 Materials provided to participants as part of the design task for recommending a menu design for room planner interface.

The participants were provided three sheets of paper as shown in figure 3-6. One with the task context, and instructions about the structure of the design advice based on the condition; a second sheet with the three menu options and descriptive information about the menu; and a third, and final, worksheet to record their recommendation. Participants had the option of either typing their recommendation or writing by hand. The three menu options were presented in random order on a single sheet of paper.

All participants received a broad overview of the purpose of the study, which was similar to the information available on the flyers. Participants were asked to read aloud the instructions on the first sheet. After they have finish reading, the researcher asked them whether they have any questions. If there were no questions, the researcher discussed the following three suggestions to the participants to help focus them on the task:
• The research is concerned with the recommendation they produce and not the correctness of the menu option they picked.
• Rank order the menu item, but that they only had to explain their top choice.
• Tag sentences with “a, b, c” in their explanation for participants belonging to the pattern or claims condition.

Participants were asked to rank the options to encourage them to actively think about their decision, and also assist them in later stages of the task. Once they had made the choice, the instructions asked them document their explanation in the worksheet provided. Time required for completing the practice task was captured separate from the design task.

3.4.5 Procedure for Expert Evaluation

Before the evaluation of the data collected from the study began, the researcher completed a training session with the evaluators. During this session, the goal of the study was explained to the expert evaluators, which involved discussing the definition for each of the metrics used to evaluate the recommendations. The evaluators each received the reference sheet shown in appendix L to consult during this evaluation and the task description (refer Appendix D). After this, each evaluator received ten recommendations that were captured during the pilot experiment. The ten recommendations consisted of four recommendations from the claims group, three from the pattern group and three from the control group. Each recommendation was presented in a separate sheet along with the metrics and the 7-pt Lickert-type rating scale (as shown in appendix M). The recommendations were randomized such that pattern, claim or control group sourced recommendations were in no particular order. To summarize the materials, each evaluator, including the researcher, received the reference sheet and a set of ten recommendations. Once this was distributed, everyone independently rated the recommendations.
on the four metrics. Following this, a debriefing session was conducted where evaluator shared their ratings for the recommendations. Each evaluator then explained, based on reference sheet, how they rated the recommendation based on the metrics. The debriefing session helped highlight the gaps in the reference sheet. As in, what more is required in the reference sheet in order to fully understand the metrics? This involved providing an example of a good recommendation and the corresponding ratings. Evaluators also asked for examples of good and poor recommendations as it related to each of the metric.

Appendix L: Reference sheet for evaluating recommendations

Quick Summary of Metrics

1. CONTENT OF USE DESCRIBED: What is the task and what worked? Why does the new design work for the task? Contrast with previous version.
2. ROBUSTNESS OF DESIGN NOISE: What are the reasons for calling this a good fit? Why not?

Metrics Explained With Examples

Looking for consistency and detail described when discussing the same option in a reference document.

Figure 3–7 Materials provided to expert evaluators consisting of the reference sheet, 52 individual pieces of advice, and one good example of advice

Answers to these clarification questions were used as a basis to make edits to the reference document show in in appendix L. The researcher conducted a second session with the
evaluators, this time with the revised reference sheet. Another set of recommendations from the pilot study was used to determine whether the metrics were clear to the evaluators. Once the evaluators confirmed that they were clear about the metrics and how to go and rate the recommendations, each evaluator received the set of recommendations from the thesis study. This amounted to 57 recommendations that were evaluated by the two expert judges independently for each of the metric present. The recommendations were randomized again such that pattern, claim and control group data was in no particular order. The expert ratings were used in the statistical analyses presented in the next chapter.
CHAPTER 4: RESULTS

This chapter reports findings from data analysis, and covers the following topics:

- Data Collection and Sample Population Overview
- Summary of Findings
- Detailed findings
  - Research Q1: Impact of advice structure on worthiness of advice measured by
    - Context of use described
    - Rationale expressed
    - Usefulness
    - Overall Quality
  - Research Q2: Impact of using advice structure on task time
- Post-task Survey Results

4.1. Introduction
This study examined the value of using a pattern or claim-derived structure on communicating interaction design advice. It has been asserted that a pattern or claim has an advantage over other forms of design advice (e.g., guidelines) because they present design advice in context (when) and include rationale (why). This assertion motivated the three research questions explored in this study. The first research question (RQ1) examined whether a pattern or claim structure had an impact on the worthiness of the design advice or on the ability to communicate design advice. This question was operationalized as four sub-questions that examined the impact of the advice structure on capturing context-of-use, rationale, usefulness, and overall quality of the advice. The second research question (RQ2) examined the impact of the using an advice structure on the time
taken to prepare design advice. And finally, the third research question (RQ3) explored whether the design advice structure (pattern or claim) influenced the design choice made.

Participants in the study were asked to adopt the role of an interaction designers instructed to design a menu for a room planner interface. The experiment task instructed the participants to explain their menu design recommendation for displaying a furniture catalog on a room planner interface from three given design alternatives. Participants were randomly assigned to one of three conditions: pattern, claim, and control. Participants assigned to pattern or claim condition received additional instructions to document their explanation or advice using a structure derived from patterns or claims respectively. The control condition did not receive any explicit structure. The data obtained from this exercise were task-time, design choice, and an explanation supporting their recommendation. Inferential statistical techniques were used, where appropriate, to answer the research questions.

The following instruments were used to obtain data from this study:

- Pre-Task Questionnaire (appendix A)
- Recommendation template (appendix F, G & H)
- Post-Task Questionnaire (appendix I & J)
- Recommendation Evaluation Template (Appendix M)

### 4.2. Participant Population Overview & Data Collection

Participants for the study were recruited from the College of Information Science and Technology (iSchool) at Drexel University. The iSchool offers both undergraduate and graduate degrees in information systems (ISYS), information technology (IT) and software engineering (SE). The graduate offerings include information science (IS) in addition to those listed before.
Participants were recruited between May 2008 and Oct 2008 from the school campus using advertisements and announcements. They were randomly assigned to one of three conditions (pattern, claim, and control) as shown in Table 4-1 below:

Table 4-1 Experimental Design showing the distribution of participants and treatment

<table>
<thead>
<tr>
<th>Condition</th>
<th>Assignment</th>
<th>N</th>
<th>Treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern</td>
<td>Random</td>
<td>19</td>
<td>Instructed to document advice using pattern-derived structure</td>
</tr>
<tr>
<td>Claim</td>
<td>Random</td>
<td>19</td>
<td>Instructed to document advice using claim-derived structure</td>
</tr>
<tr>
<td>Control</td>
<td>Random</td>
<td>19</td>
<td>No structure</td>
</tr>
</tbody>
</table>

All study participants were undergraduate students pursuing a bachelor's degree. A majority of participants were majoring in Information Systems & Information Technology, as shown in Table 4-2.

Table 4-2 Distribution of participants' educational background and age

<table>
<thead>
<tr>
<th>Major (Bachelors)</th>
<th>Age</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>18-25</td>
<td>26-35</td>
</tr>
<tr>
<td>Computer Engineering</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Information Systems</td>
<td>38</td>
<td>3</td>
</tr>
<tr>
<td>Information Technology</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>Software Engineering</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>53</td>
<td>3</td>
</tr>
</tbody>
</table>
Prior experience in HCI was a pre-requisite for the study. This criterion was operationalized as having completed prior coursework in HCI. Participants were expected to have completed at least an introductory course in HCI thus granting them familiarity with concepts like design guidelines or user-based design practices. Table 4-3 shows that there were almost an equal number of participants who had completed at least one or two courses in HCI. A majority of participants also reported to have at least 0-6 months of professional experience in interaction design either through jobs held or internships.

Table 4-3 Distribution of participants’ experience working in interaction design and number of courses completed in HCI

<table>
<thead>
<tr>
<th>Experience (months)</th>
<th>No of HCI Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>One</td>
</tr>
<tr>
<td>0-6</td>
<td>23</td>
</tr>
<tr>
<td>7-12</td>
<td>3</td>
</tr>
<tr>
<td>12-24</td>
<td>0</td>
</tr>
<tr>
<td>36-60</td>
<td>0</td>
</tr>
<tr>
<td>60+</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>27</td>
</tr>
</tbody>
</table>

Participants in the study where first asked to provide some demographic information (e.g., coursework, experience), which was collected using a pre-task questionnaire (Appendix A). After completing their pre-task questionnaire, participants were given a practice design problem to familiarize them with the test set-up, and to introduce them to the structure they will use for
preparing recommendations. For the practice task, they were asked to make a design choice among three design alternatives for an icon-selection task to be used on a mobile phone interface. Participants were asked to read the given instructions on their own. They were asked whether they understood the design task, and if they had any questions about the practice task. Participants were then given time to make a selection, following which they were interviewed briefly about their design choice. The interview was conducted using a script that reflected the experimental condition they were assigned (i.e., pattern, claim, control) as shown in table 4-4. At no point during the practice-task interview were the participants coaxed to give the “right answers.” It was made clear to them that there is no one right answer or design choice, and all that was expected of them was to provide their reasons for picking a particular choice. There was no time limit for this practice task, but the exercise was timed to check later whether too much or too little time was spent explaining across conditions. This practice task was marked complete when the participants said they had nothing more to add to their design advice.

Table 4-4 Additional instructions provided to the participants based on the assigned condition

<table>
<thead>
<tr>
<th>Pattern Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the user trying to do?</td>
</tr>
<tr>
<td>How will the recommended menu option be used?</td>
</tr>
<tr>
<td>Explain why it's the best option in this given situation?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Claims Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>How will the recommended menu option be used?</td>
</tr>
<tr>
<td>Advantages of using this design choice?</td>
</tr>
<tr>
<td>Disadvantages of using this design choice?</td>
</tr>
</tbody>
</table>
106

**Control Condition**

Explain why you picked a particular design choice.

For the experiment task, each participant was again asked to adopt the role of a designer and recommend a menu design for a room planner interface. Participants received a stripped down version of room-planner (appendix B) on paper. The instructions defined a room planner would enable users to add and remove furniture from an on-line catalog to a room plan, and allow them to try out different arrangements. They were also provided with three design alternatives (menus), like in the practice task, shown side-by-side on a separate sheet of paper. This menu options sheet included a brief description of the menu attributes. Participants were told that there was no one right choice. The instructions they received asked them to complete two activities. First, to rank order the three menu options in their order of preference. And secondly, to type out their recommendation using the advice structure shown in table 4-4 for the menu option they recommended. The recommendation template was used to collect this information. They were given as much time as needed to review the instructions and to ask clarification questions. Both, the time required for reviewing the instructions, and time taken to complete typing up their recommendation was recorded. All participants were asked to explain their design decision as if they were teaching someone else to make a similar choice, and that others reviewing this design advice may not see the menu options they rejected.

On an average, participants took 2.73 minutes from completing the practice task. They took 2.92 minutes for reviewing the instructions, and 9.43 minutes for completing the design task. Table 4-5 shows the average time taken for completing each phase for each experiment condition.
Table 4-5 Average time (in minutes) taken to complete the practice and design task for each experiment condition

<table>
<thead>
<tr>
<th></th>
<th>Practice task</th>
<th>Instructions</th>
<th>Writing</th>
<th>Overall (Total)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern</td>
<td>2.86</td>
<td>2.86</td>
<td>9.87</td>
<td>15.59</td>
</tr>
<tr>
<td>Claim</td>
<td>3.19</td>
<td>3.30</td>
<td>9.86</td>
<td>16.35</td>
</tr>
<tr>
<td>Control</td>
<td>2.14</td>
<td>2.60</td>
<td>8.55</td>
<td>13.29</td>
</tr>
</tbody>
</table>

Once the participants had completed the design task, they filled out a post-task questionnaire (Appendix I or Appendix J for control group). This questionnaire was used to collect feedback on their experiences during the study, and how they rated the usefulness of structure used to articulate advice. Participants were compensated $15 for their effort.

### 4.3 Summary of Results

Table 4-6 shows a summary of the results for each research question along with the statistical tests used to evaluate the differences between the conditions. Following sections describe the findings in more detailed, organized by research question.

<table>
<thead>
<tr>
<th>Research Question</th>
<th>Results</th>
<th>Statistical Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ 1</td>
<td>No significant difference</td>
<td>ANOVA</td>
</tr>
<tr>
<td>Context of Use</td>
<td>No significant difference</td>
<td>ANOVA</td>
</tr>
<tr>
<td>Rationale</td>
<td>No significant difference</td>
<td>ANOVA</td>
</tr>
<tr>
<td>Usefulness</td>
<td>Significant difference, p &lt; .0125</td>
<td>ANOVA</td>
</tr>
</tbody>
</table>
4.4 RQ I: Impact on the effectiveness of Design Advice

To determine the “goodness” of design advice, the study used expert judges to rate each advice on the following four criteria, framed by the four sub-research questions:

- To what extent does the structure used for design advice impact the context of use described?
- To what extent does the structure used for design advice affect the rationale expressed?
- To what extent does the structure used for design advice impact the usefulness of the advice?
- To what extent does the structure used for design advice impact the overall quality of advice?

Expert judging has been used in previous empirical studies on patterns, and hence decided to be an appropriate technique for evaluating design advice (narratives). The study used two experts, both professors in human-computer interaction. Each judge had at least 15-20 years of experience teaching and conducting research on the design and evaluation of interactive systems.
Before rating the advice prepared by the participants, a training session was scheduled with the judges. During the training session the judges were introduced them to an evaluation template that used 7-pt Lickert type scales for rating the data (c.f. appendix L). The judges first independently rated data collected from a pilot study to gain familiarity with the evaluation metrics and also the type of data (narratives) they could expect from the experiment. After this training session, a meeting was held between the researcher and the judges to clarify and reconcile any differences in how the pilot data was rated. An independent rating of the experiment data followed this session. In this study, the data obtained from the rating scales were assumed as interval-type and not ordinal. Further analyses on the ratings data were conducted with this assumption.
4.3.1 Assessing Inter-rater Reliability

Intraclass correlation in SPSS was used as a measure of assessing inter-rater reliability. ICC (3, k) has been suggested in the literature [Shrout and Fleiss, 1996] when each target is rated independently by a fixed number of judges (i.e., a two-way mixed model, where k=2). Since the ratings were averaged, the consistency option was used instead of perfect agreement.

<table>
<thead>
<tr>
<th>For Metric 1</th>
<th>Rater 1</th>
<th>Rater 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Case 2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Case 3</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Case 4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Case 5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Case 6</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Figure 4-2 Illustrating an example rating set where the judges were consistent, but not in agreement.

Instead of looking at whether both judge agree on the same rating, consistency looks at whether the ratings went up or down proportionally from case to case. In other words, looking at the figure, as long as the slopes remain parallel between any two cases, it would indicate good consistency even when judges don’t agree on a particular rating (c.f. figure 4-2). Consistency accounts for the fact that it may be difficult for two independent raters to perfectly agree on ratings. The narrative nature of data made this even more challenging. Given these constraints,
consistency was a better approach. For the experiment ratings, the judges displayed good consistency in their ratings. Intraclass correlation coefficient for average measures = 0.815 (Cronbach’s Alpha). Further parametric analyses used an average of the ratings by the two judges.

Table 4-7 Intra-class correlation for individual metrics and for the entire sample

<table>
<thead>
<tr>
<th></th>
<th>Intraclass Correlation (a)</th>
<th>95% Confidence Interval</th>
<th>F Test with True Value 0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lower Bound</td>
<td>Upper Bound</td>
<td>Value</td>
</tr>
<tr>
<td>Single Measures</td>
<td>.687</td>
<td>.481</td>
<td>.821</td>
</tr>
<tr>
<td>Average Measures</td>
<td>.815</td>
<td>.649</td>
<td>.902</td>
</tr>
</tbody>
</table>

4.3.2 Exploring Data: Normality and Equal Variances

The descriptive statistics for the dependent variables are shown in table 4-8, which shows mean, standard deviation, and median.

Table 4-8 Mean, standard deviation and median for the dependent variables (ratings)

<table>
<thead>
<tr>
<th>Dependent Variable</th>
<th>Mean</th>
<th>SD</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context of Use</td>
<td>2.80</td>
<td>1.05</td>
<td>3.00</td>
</tr>
<tr>
<td>Rationale</td>
<td>3.58</td>
<td>1.10</td>
<td>3.50</td>
</tr>
<tr>
<td>Useful</td>
<td>3.30</td>
<td>1.09</td>
<td>3.5</td>
</tr>
<tr>
<td>Overall Quality</td>
<td>3.18</td>
<td>0.86</td>
<td>3.50</td>
</tr>
</tbody>
</table>

Skewness and kurtosis value for each dependent variable under worthiness was calculated to assess normality. Skewness checks whether the sample distribution is symmetrical, and kurtosis
gives a measure of the “peakiness” of the distribution. In theory, for a normal distribution the skewness and kurtosis values are zero, but in experiments it’s common to find sample distributions that are slightly skewed, and also displaying a non-zero value for kurtosis. A higher positive value for kurtosis indicates a pronounced peak (higher frequency of cases with a common value clustering around the mean), whereas negative value indicates a more flat distribution or spread. Similarly, negative value for skewness indicates that the distribution is skewed right and vice-versa for a positive value. Table 4-9 shows the skewness and kurtosis value for the sample distribution, along with their respective standard error.

Table 4-9 Checking for normality for context of use, rationale expressed, usefulness and overall quality before conducting statistical tests (ANOVA)

<table>
<thead>
<tr>
<th>Dependent Variable</th>
<th>Skewness</th>
<th>Std. Error Skewness</th>
<th>Kurtosis</th>
<th>Std. Error Kurtosis</th>
<th>Mean</th>
<th>5% Trimmed Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context of Use</td>
<td>+ 0.58</td>
<td>0.316</td>
<td>+ 0.87</td>
<td>0.623</td>
<td>2.80</td>
<td>2.75</td>
</tr>
<tr>
<td>Rationale</td>
<td>+ 0.05</td>
<td></td>
<td>- 0.67</td>
<td></td>
<td>3.58</td>
<td>3.57</td>
</tr>
<tr>
<td>Useful</td>
<td>+ 0.28</td>
<td></td>
<td>- 0.87</td>
<td></td>
<td>3.30</td>
<td>3.26</td>
</tr>
<tr>
<td>Overall Quality</td>
<td>- 0.16</td>
<td></td>
<td>- 1.28</td>
<td></td>
<td>3.18</td>
<td>3.18</td>
</tr>
</tbody>
</table>

The skewness and kurtosis values reveal that the distribution for context, rationale and usefulness is slightly skewed left, whereas quality is skewed right. A 5% trimmed mean technique was used to check for outliers (i.e., by removing the top and bottom 5% of the cases and recalculating the
mean). Table 4-9 shows that the trimmed mean is similar to the sample mean, and hence it is safe to assume that the data is normal.

4.3.3 Relationship between Dependent Variables

SPSS correlate function was used to determine the correlations between the dependent variables. Pearson’s correlation coefficient (r) was used instead of Spearman’s based on the assumption that the rating data were interval. Table 4-10 shows the correlation coefficients and variance explained between any two dependent variables. The three measures commonly used to interpret correlations are significance, strength, and directionality. Significant correlation indicates that the relationship between any two variables is not based on chance. The strength of correlation (R or r²) can be stated as the amount by which one variable explains the variance of another. A negative sign in front of the correlation coefficient indicates that the two variables are inversely related.

Table 4-10 Correlation coefficients, significance, and strength of correlation between any two dependent variables

<table>
<thead>
<tr>
<th>Dependent Variable</th>
<th>Coefficient and Strength</th>
<th>Context</th>
<th>Usefulness</th>
<th>Rationale</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context</td>
<td>Pearson Correlation</td>
<td>--</td>
<td>.407**</td>
<td>.465**</td>
<td>.701**</td>
</tr>
<tr>
<td></td>
<td>Variance Explained</td>
<td>--</td>
<td>16.5%</td>
<td>21.6%</td>
<td>49.1%</td>
</tr>
<tr>
<td>Usefulness</td>
<td>Pearson Correlation</td>
<td>.407**</td>
<td>--</td>
<td>.743**</td>
<td>.815**</td>
</tr>
<tr>
<td></td>
<td>Variance Explained</td>
<td>16.5%</td>
<td>--</td>
<td>55.2%</td>
<td>66.4%</td>
</tr>
<tr>
<td>Rationale</td>
<td>Pearson Correlation</td>
<td>.465**</td>
<td>.743**</td>
<td>--</td>
<td>.843**</td>
</tr>
<tr>
<td></td>
<td>Variance Explained</td>
<td>21.6%</td>
<td>55.2%</td>
<td>--</td>
<td>71%</td>
</tr>
</tbody>
</table>
Table 4-10 shows that some dependent variables are highly correlated, whereas some are not (based on variance explained). All variables have a significant correlation with each other. Usefulness and Rationale were highly correlated (positive) with quality explaining for 66.4% and 71% of the variance respectively. Context explained 49% of the variance in quality ratings. Usefulness explained 55% of the variance in rationale ratings.

<table>
<thead>
<tr>
<th>Quality</th>
<th>Pearson Correlation</th>
<th>Variance Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>.701**</td>
<td>49.1%</td>
</tr>
<tr>
<td></td>
<td>.815**</td>
<td>66.4%</td>
</tr>
<tr>
<td></td>
<td>.843**</td>
<td>71%</td>
</tr>
</tbody>
</table>

** Significant at p<.01 level

When the variables are moderately correlated (0.2 < r < 0.8), as in the case here, textbooks (Pallant, 2007; Tabachnick & Fidell, 2000) suggest using MANOVA as the omnibus test for comparing group means instead of multiple ANOVAs so as to avoid type-I error. Type-I error is when the null hypothesis is rejected when it should not be. Moderate correlation implies that some linear combination of dependent variables is possible, which ANOVA ignores. One drawback or benefit in using MANOVA is that while the chance of type-I error is reduced, the variable compared is a vector (linear combination of dependent variables). ANOVA is still appropriate if the goal is to examine the mean differences for each dependent variable. Both Pallant (2007) and Tabachnick & Fidell (2000) suggest using Bonferroni’s correction to reduce the chance of type-I error when running multiple ANOVAs. Bonferroni’s correction involves setting a conservative number for assessing significance. This number is obtained by dividing the alpha value of 0.05 by the number of dependent variables or number of ANOVA tests planned. In this study, the new alpha value of .0125 (.05/4) was used to check for significant differences in the post-hoc test. Tukey’s post-hoc test is known to account for the type-I error well (Fields,
2005). The trade-off of using this correction is that a more conservative alpha level increases the chance of type-II error, where null hypothesis is assumed to be true when it is not.

### 4.3.4 Comparing Means Across Conditions

This section reports the findings from ANOVA and post-hoc tests, when appropriate, for the four dependent variables separately.

#### 4.3.4.1 Context Of Use Described

A one-way between subjects analysis of variance was completed to compare the effect of structure (independent variable) on the context of use described. The independent variable had three levels: pattern, claim, and control. The participants in the pattern condition had received instructions to prepare their recommendation using a structure derived from patterns. Similarly, participants in the claim condition received a structure derived from a claim. The control group did not receive any explicit structure. Levene’s test for equality for variances did not find any significant differences, and hence the data does not violate the assumptions for ANOVA.

ANOVA found no statistically significant difference at $p < .0125$ for mean rating for context of use among the three groups $F (2, 54) = .52$, $p = .596$
Figure 4–3 Means plot for Context of Use Described

Table 4-11 Descriptive Statistics for Context of Use Described

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>Mean</th>
<th>Std. Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pattern</td>
<td>19</td>
<td>3.000</td>
<td>1.06719</td>
</tr>
<tr>
<td>Claims</td>
<td>19</td>
<td>2.684</td>
<td>1.13297</td>
</tr>
<tr>
<td>Control</td>
<td>19</td>
<td>2.710</td>
<td>.96200</td>
</tr>
<tr>
<td>Total</td>
<td>57</td>
<td>2.798</td>
<td>1.04736</td>
</tr>
</tbody>
</table>

4.3.4.2 Rationale Expressed
A one-way between subjects analysis of variance was completed to explore the effect of structure (independent variable) on rationale expressed or how well the participants provided arguments for selecting a design option. The independent variable had three levels: pattern, claim, and control. Levene’s test for equal variances did not find any significant differences, and hence data does not violate the assumptions for ANOVA. There was no statistically significant difference at \( p<.0125 \) for mean rating for rationale among the three groups \( F (2, 54)=.859, p=.429 \)

![Means plot for Rationale](image)

**Figure 4–4** Means plot for Rationale

**Table 4-12** Descriptive Statistics for Rationale Expressed

<table>
<thead>
<tr>
<th>Condition</th>
<th>N</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Claim</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.3.4.3 Usefulness

A one-way between subjects analysis of variance was conducted to compare the effect of structure (independent variable) on the usefulness of advice. The advice was deemed useful if it included a discussion of implications-on-use relevant to the design task. In other words, it measured whether the rationale expressed was relevant. The independent variable had three levels: pattern, claim, and control. Levene’s test for equal variances did not find any significant differences, and hence data does not violate the assumptions for ANOVA. There was a statistically significant difference at \( p < .0125 \) for mean rating for usefulness among the three groups \( F(2, 54) = 6.78, p = .002 \). Tukey post-hoc test showed that the difference between means for claim-pattern and claim-control was significant. Mean difference between pattern-claim was not significant.
Figure 4–5 Means plot Usefulness

Table 4-13 Descriptive Statistics for Usefulness

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usefulness</td>
<td>19</td>
<td>3.1316</td>
<td>.99780</td>
<td>5.00</td>
</tr>
<tr>
<td>Claims</td>
<td>19</td>
<td>3.9474</td>
<td>1.11673</td>
<td>5.50</td>
</tr>
<tr>
<td>Control</td>
<td>19</td>
<td>2.7895</td>
<td>.85498</td>
<td>4.50</td>
</tr>
<tr>
<td>Total</td>
<td>57</td>
<td>3.2895</td>
<td>1.09360</td>
<td>5.50</td>
</tr>
</tbody>
</table>

4.3.4.4 Overall Quality

A one-way between subjects analysis of variance was computed to compare the effect of structure (independent variable) on the overall quality of advice. The independent variable had three levels:
pattern, claim, and control. Levene’s test for equal variances did not find any significant
differences, and hence data does not violate the assumptions for ANOVA. There was no
statistically significant difference at $p < .0125$ for mean rating for quality among the three groups
$F (2, 54) = .74$, $p = .48$.

Figure 4–6 Means plot for Overall Quality

Table 4-14 Descriptive Statistics for Overall Quality

<table>
<thead>
<tr>
<th>Quality</th>
<th>N</th>
<th>Mean</th>
<th>Std. Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern</td>
<td>19</td>
<td>3.182</td>
<td>.86940</td>
</tr>
<tr>
<td>Claims</td>
<td>19</td>
<td>3.342</td>
<td>.89834</td>
</tr>
</tbody>
</table>
### Summary of Findings for Worthiness of Design Advice

Comparison of means for context of use, rationale, usefulness, and overall quality yielded the following findings:

- Participants using the *claim* structure prepared design advise that was judged to be more useful.
- Advice prepared using the *claim* structure was also rated higher for *rationale* expressed and *overall quality*; however, the mean differences were not statistically significant.
- Only for *context of use*, the expert judges rated advice prepared using the *pattern* structure higher, but the difference between means was not statistically significant.
- For *usefulness*, the mean rating for the *claim* condition was a 26% improvement over *pattern* group, and a 41% improvement over the control condition.
- The mean rating for the *pattern* condition for *context-of-use*-described was slightly higher, an approx. 12% and 11% improvement over mean ratings received by *claim* and *control* condition respectively.
- **Correlations**:
  - Usefulness and Rationale were highly correlated (positive) with quality, explaining for 66.4% and 71% of the variance in quality rating respectively.
  - Context explained 49% of the variance in quality ratings.
  - Usefulness explained 55% of the variance in rationale ratings.
Based on this analysis, it appears that pattern structure influenced how participants discussed context of use, but only marginally. The claim structure was more successful in eliciting advice that seemed more useful.

Quantitative analysis discussed whether the groups were different, but without inspecting the advice itself, little can be said about why a particular advice may have received higher or lower ratings from the judges. In order to better understand this, the study includes a content-analysis approach to explore the differences and affinities present in advice prepared using the different structures. Since this is not part of the primary analysis, and being more explorative in nature, the findings from content analysis are presented in the section titled Additional Findings. Content analysis serves the following two purposes: First, it provides a more detailed analysis of design advice, seeking evidence for context of use, rationale and usefulness; and secondly, it looks for some form of corroboration for the expert ratings (e.g. correlations).

4.5 RQ II: Impact of Advice Structure on Design Time

This section reports on the impact of the advice structure on the time taken to prepare design advice. The advice structure used was the independent variable with three levels: pattern, claim and control. Participants were randomly assigned to one of the three conditions. Time (dependent variable) was recorded for three activities: practice task, instructions, and writing design advice. This study considered the time required to review the instructions and time required to complete writing the advice as the total time for completing the experiment task referred to as design time. Table 4-15 shows the descriptive statistics for design time spent during the experiment and the tests for assessing normality.
Table 4-15 Assessing normality for design time

<table>
<thead>
<tr>
<th>Design Time</th>
<th>Statistic</th>
<th>Std. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>12.3447</td>
<td>.69756</td>
</tr>
<tr>
<td>5% Trimmed Mean</td>
<td>12.0567</td>
<td></td>
</tr>
<tr>
<td>Median</td>
<td>12.0000</td>
<td></td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>5.26643</td>
<td></td>
</tr>
<tr>
<td>Skewness</td>
<td>.668</td>
<td>.316</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>.230</td>
<td>.623</td>
</tr>
</tbody>
</table>

The data shows that the distribution can be assumed normal, but is slightly skewed to the left.

Table 4-16 shows the descriptive statistics for time spent during the practice time and time taken to complete the exercise (design time).

Table 4-16 Time taken to complete the experiment

<table>
<thead>
<tr>
<th></th>
<th>Practice Time (mins)</th>
<th>Design Time (mins)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std. Dev</td>
</tr>
<tr>
<td><strong>Pattern</strong></td>
<td>2.86</td>
<td>0.946</td>
</tr>
<tr>
<td><strong>Claim</strong></td>
<td>3.19</td>
<td>0.866</td>
</tr>
<tr>
<td><strong>Control</strong></td>
<td>2.14</td>
<td>0.799</td>
</tr>
</tbody>
</table>

Levene’s test for design time was not significant, so the equal variances assumption for ANOVA was not violated. A one-way between subjects analysis of variance was conducted to compare the
impact of structure (independent variable) on the time required to prepare design advice (design time). The independent variable had three levels: pattern, claim, and control. There was no statistically significant difference at $p < .05$ level for mean design time among the three experiment conditions $F (2, 54) = .76$, $p = .44$.

![Estimated Marginal Means of Design Time](image)

Figure 4–7 Means plot for Design Time.

The difference between the conditions for design time was marginal with claims conditions requiring slightly longer time to complete the task. The relationship between worthiness of design advice (RQ1) and design time (RQ2) was explored using correlation.
Table 4-17 Assessing correlation between design time and dependent variables in research question 1.

<table>
<thead>
<tr>
<th></th>
<th>Context</th>
<th>Rationale</th>
<th>Usefulness</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design_Time Pearson</td>
<td>Correlation</td>
<td>.171</td>
<td>.357**</td>
<td>.278*</td>
</tr>
<tr>
<td></td>
<td>Sig. (2-tailed)</td>
<td>.202</td>
<td>.006</td>
<td>.036</td>
</tr>
<tr>
<td></td>
<td>Variance Explained</td>
<td>2.9%</td>
<td>12.74%</td>
<td>7.7%</td>
</tr>
</tbody>
</table>

**: Correlation is significant at the 0.01 level (2-tailed).

*: Correlation is significant at the 0.05 level (2-tailed).

The correlations between design time, rationale, usefulness and overall quality were found to be statistically significant, but explained a very small percentage of variance. Design time was weakly correlated with the dependent variables from in the first research question. Context of use described displayed the weakest correlation with design time and was not statistically significant.

In summary, the structure used for preparing design advice had no impact on the time taken to complete the exercise. Design time was found to be weakly correlated with the other dependent variables in RQ1.

### 4.6 Additional Findings from Content-Analysis

This section attempts to reconcile findings from two sources, one obtained from expert judging and the other from a quantitative content analysis of design advice. Content analysis, simply
explained, is an analysis of the actual message content (e.g., video, text, pictures) to piece together a richer picture of what, why and how the message was communicated (Strijbos et al., 2006). The analysis presented in this section is based on examining design advice using a coding scheme. The coding scheme was derived from the dependent variables used by expert judges to rate design advice (Appendix L). The following codes were created to reflect aspects commonly used in describing context of use: user, task, scenario, examples, and spatial. Instead of looking for references in the design advice that may correspond to context directly. Similarly, rationale was coded as evaluative comments (both positive and negative) made by participants about the menu, and usefulness was coded as design issues addressed. Appendix L shows the glossary of codes used to guide content analysis.

Strijbos et al. (2006) provide a simple introduction to the different approaches adopted in content analysis, namely quantitative and qualitative. To briefly illustrate the differences between the two approaches, consider this example (adapted from Strijbos, et al. 2006). In a virtual community (e.g., community message boards), counting the number of messages exchanged between any two members may be considered as a measure for the degree of participation. Such an analysis at some level may yield a possible structure for articulating relationships, a form of network analysis, to visualize who talks to whom. A slightly more detailed analysis may also be attempted to look at the content of messages to determine what kind of communication was exchanged (e.g., work, casual, meetings). The value of this analysis tends to depend on the research question being explored. That is, whether the study was for looking evidence to support hypotheses or answer a set of research questions and provide a richer evidence describing what, why and/or how an online community behaves.
The coding scheme was developed in top-down or deductive manner, where it was derived from the definition of dependent variables provided to the judges. The alternative is create the codes inductively from the data. While the two approaches seem like two extremes to coding approaches (or content analysis), it may be difficult to argue whether either code generation approach was pure induction or apriori. The objective in both cases is to present evidence to answer research questions, and to exercise rigor when conducting the analysis. Like the example described above, this study approached content analysis from quantitative perspective, but went into more detail than “message length.” Message length in this situation was considered as the number of words used in the advice.

Summary of findings for RQ1 show that advice prepared using a claim structure was rated as more USEFUL, and possibly expressed RATIONALE better. Both these variables were also found positively correlated with the OVERALL QUALITY ratings. The advice prepared using the pattern structure seemed to relatively address CONTEXT better than the other structures, but not by a significant margin. The goal of this analysis is to offer a possible explanation for why some advice was rated higher than others. Quality was not examined because it was highly correlated with CONTEXT-OF-USE (CoU), RATIONALE, and USEFULNESS, which implied that if an advice scored highly on these three variables, it may also imply higher quality.

4.6.1 Summary of Findings from Content Analysis
Table 4-18 provides an overview of the analyses of measures derived from content analysis. It explores the following questions:

- Did the advice structure affect how much description was captured, overall?
- To what extent did the advice structure influence how context of use was described?
• Did the advice structure influence reflection on design decisions, measured by number of evaluative comments made?
• Did the advice structure affect how many unique design issues or implications were identified?

Table 4.18 Summary of findings from Content Analysis. Significance at p<.05 level

<table>
<thead>
<tr>
<th>Additional Analyses</th>
<th>ANOVA Results</th>
<th>Comparing Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word Count</td>
<td>Significant Difference</td>
<td>Pattern &amp; Claim &gt; Control</td>
</tr>
<tr>
<td>Context_Count</td>
<td>Significant Difference</td>
<td>Pattern &gt; Control</td>
</tr>
<tr>
<td>No. of Evaluative Comments</td>
<td>Significant Difference</td>
<td>Claim &gt; Pattern &amp; Control</td>
</tr>
<tr>
<td>No. of Unique Design Issues</td>
<td>Significant Difference</td>
<td>Claim &gt; Pattern &amp; Control</td>
</tr>
</tbody>
</table>

4.6.2 Impact of Advice Structure on Number of Words in Advice

A one-way between subjects analysis of variance was conducted to compare the effect of structure (independent variable) on the number of words (#Words) present in the advice. The independent variable had three levels: pattern, claim, and control. The participants in the pattern condition had received instructions to prepare their recommendation using a structure derived from patterns. Participants in the claim condition received a structure derived from a claim. The control group did not receive any explicit structure. Table 4-19 shows the descriptive statistics for #Words along with tests for assessing normality.

Table 4-19 Average number of words used to describe design decision and its standard deviation
<table>
<thead>
<tr>
<th></th>
<th>Value 1</th>
<th>Value 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>181.3158</td>
<td>9.75358</td>
</tr>
<tr>
<td>5% Trimmed Mean</td>
<td>179.5741</td>
<td></td>
</tr>
<tr>
<td>Median</td>
<td>187.0000</td>
<td></td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>73.63791</td>
<td></td>
</tr>
<tr>
<td>Skewness</td>
<td>.173</td>
<td>.316</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>-.923</td>
<td>.623</td>
</tr>
</tbody>
</table>

The participants used a mean of 181.3 words to explain their design decision. The distribution was slightly skewed to the left and flatter than a theoretical normal distribution. The 5% trimmed mean technique, used to check for the influence of outliers, did not vary the mean by much. Based on this it was safe to assume the underlying distribution of word counts as normal.

Levene’s test for equality of variances did not find any significant differences, and hence the data did not violate the assumptions for ANOVA. A one-way between group analysis of variance found a statistically significant difference at p<.05 for mean #words among the three groups F (2, 54)=4.27, p=.02. Tukey post-hoc test showed that both pattern and claim condition used more words to describe design advice compared to the control group. The mean difference between the pattern and claim condition for #words was not significant.
Figure 4–8 Means plot for number of words (#words) present in design advice

Table 4-20 Means and Standard Deviation for #Words used in the three conditions.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern</td>
<td>200.3684</td>
<td>78.09838</td>
<td>19</td>
</tr>
<tr>
<td>Claims</td>
<td>200.4211</td>
<td>72.79752</td>
<td>19</td>
</tr>
<tr>
<td>Control</td>
<td>143.1579</td>
<td>56.26649</td>
<td>19</td>
</tr>
<tr>
<td>Total</td>
<td>181.3158</td>
<td>73.63791</td>
<td>57</td>
</tr>
</tbody>
</table>

The dependent variable, #Words, was positively and highly correlated with the ratings received from judges. The #words present in the advice explained almost 65% of the variance in quality.
ratings. All the correlations were statistically significant at \( p < 0.01 \) level. The \#words was also highly correlated with usefulness and rationale ratings. Table 4-21 shows the correlation coefficients and strength of correlation between the dependent variable and measures for worthiness of advice.

Table 4-21 Correlations between the measures for worthiness of advice and number of words present in the design advice.

<table>
<thead>
<tr>
<th>Word Count</th>
<th>Context</th>
<th>Usefulness</th>
<th>Rationale</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pearson Correlation</td>
<td>.548**</td>
<td>.741**</td>
<td>.715**</td>
<td>.808**</td>
</tr>
<tr>
<td>Variance Explained</td>
<td>30.03%</td>
<td>54.9%</td>
<td>51.12%</td>
<td>65.28%</td>
</tr>
</tbody>
</table>

**. Correlation is significant at the 0.01 level (2-tailed).

4.6.3 Effect of Advice Structure on Context of Use Described

This section reports the impact of advice structure on how much participants discussed the context of use. The following codes were used to identify context of use:

- **Scenario** code was applied in situations where the design advice presented a usage scenario. A scenario usually included a discussion of design features embedded in sequence of events that were in a logical and temporal sequence. For example, the user will click on the drop-down arrow to reveal the categories and then will select the category “tables.” They will use the scroll bar to navigate to a preferred item, and drag-drop it into the room plan.

- **Task** code identified references where the advice described how a user might use a design feature to accomplish a sub-goal related to furniture shopping. A task was stated
without indicating any temporal sequence of events thus showing only a fragment of scenario. Neither were task references were preceded or followed by a related sub-goal. The scenario code and task code were mutually exclusive. For example, multiple categories can be opened to compare furniture items across categories (in a directory-tree menu).

- **User** code was applied to references when the design advice discussed the implication of a design feature in terms of its effect on the user or user goals. For example, words like confusing, cluttered or over-whelming implied a consideration of the users.

- **Examples** code was applied when participants made specific reference to products or artifacts related to the task domain to make their description more concrete. For example referring to products as sofas or lamps instead of calling them just products.

- **Spatial** code was applied when the advice referred to spatial aspects of the room planner. For example discussing a fixed menu height or discussing the number of menu items that can be displayed.

Context_Count was defined as a dependent measure that indicated the word count specific to discussion of context, as determined by the codes. As stated above, while the task and scenario codes were exclusive, examples and user codes were not. This meant that a reference coded for example may overlap with references codes as a scenario in terms of coverage. However, the word count for context (context_count) took this into account, and did not count the reference twice because two codes overlapped. That is, if a scenario description comprised of 200 words, and an example code was applied to an embedded reference amounting to 50 words, the context_count (scenario+example) was still counted as 200 words and not 250. Table 4-22 below shows the descriptive statistics for words counted under context along with the two tests for determining normality: skewness/kurtosis and 5% trimmed means.
Based on the tests, it was determined that context_count did not violate the assumptions for normality, and it was safe to assume that the underlying distribution was normal.

Table 4-22 Mean and Standard deviation for Context_Count and checking for Normality.

<table>
<thead>
<tr>
<th>Context_Count</th>
<th>Statistic</th>
<th>Std. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>133.1404</td>
<td>9.79212</td>
</tr>
<tr>
<td>5% Trimmed Mean</td>
<td>133.0019</td>
<td></td>
</tr>
<tr>
<td>Median</td>
<td>135.0000</td>
<td></td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>73.92888</td>
<td></td>
</tr>
<tr>
<td>Skewness</td>
<td>.057</td>
<td>.316</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>-.845</td>
<td>.623</td>
</tr>
</tbody>
</table>

A one way between groups analysis of variance was conducted to see whether the independent variable (advice structure) influenced how much participants discussed context-of-use, measured by context_count. The independent variable had the following three levels: pattern, claim, and control. The Levene’s test for equal variances (HOV) found a significant difference, p = .041, strictly violating one of the assumptions for ANOVA. Pallant (2004) suggests that since ANOVA is a relatively robust technique, its resilient to violation of the equal variances assumption provided the group sizes are similar, as was the case. Therefore, ANOVA was still considered an appropriate technique.
ANOVA found the mean difference in context_count to be significant at the $p < .05$ level, $F(2, 54) = 4.74, p = .013$. Tukey post-hoc test confirmed that context_count for the pattern condition was significantly higher than the control condition. The mean differences in context_count between the pattern and claim condition was not statistically significant.

![Means plot for Context_Count (number of words coded describing context)](image)

Figure 4–9 Means plot for Context_Count (number of words coded describing context)

<table>
<thead>
<tr>
<th>Condition</th>
<th>N</th>
<th>Mean</th>
<th>Std. Deviation</th>
</tr>
</thead>
</table>

Table 4-23 Mean and Standard Deviation for Context_Count
To explore the relationship between how much design advice discussed context-of-use and the rating received for context, a Pearson's correlation was computed between the two. There was a positive and strong correlation between context_count and context-of-use rating, significant at p < .05 level. The correlations between context_count and other dependent measures in RQ1 were also statistically significant as shown in table 4-24. Context_count accounted for 25.9% of variance in context ratings.

Table 4-24 Correlation between Context_Count and Context-of-Use Rating

<table>
<thead>
<tr>
<th>Context_Count</th>
<th>Context</th>
<th>Rationale</th>
<th>Usefulness</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pearson Correlation</td>
<td>.509**</td>
<td>.405**</td>
<td>.484**</td>
<td>.608**</td>
</tr>
<tr>
<td>Sig. (2-tailed)</td>
<td>.000</td>
<td>.002</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>Variance Explained</td>
<td>25.9%</td>
<td>16.4%</td>
<td>23.4%</td>
<td>36.9%</td>
</tr>
</tbody>
</table>

**. Correlation is significant at the 0.01 level (2-tailed).

### 4.6.4 Impact of Advice Structure on Evaluative Comments Made

This section reports findings on the impact of advice structure on the number of evaluative comments (EvalComment) made. The independent variable had three levels: pattern, control, and
control. The dependent measure was the number of evaluative comments. Evaluative comments included both positive (+ve) and negative (-ve) evaluative comments. A positive evaluative comment was identified as a reference describing a positive implication of a design feature. A reference that described a negative implication of a design feature was coded as a negative evaluative comment. The total number of evaluative comments was a total of positive and negative comments. The positive and negative EvalComment codes (or references) were mutually exclusive.

Example of a +ve comment: “The drop-down is a better feature because it focuses the user to pay attention to one category at a time without distracting them with too many options.”

Example of a -ve comment: “The + and – sign on the category label may not be very intuitive to the user.”

Table 4-25 reports the means and standard deviation for number of evaluative comments. Skewness and Kurtosis values were computed to determine whether the distribution of comments made fit a normal distribution. The distribution was slightly skewed to the left, and kurtosis value showed that the distribution had a pronounced peak. The 5% trimmed means technique computes the mean after removing the top and bottom 5% of the cases to account for outliers. Inspection using the 5% trimmed means technique did not reveal a large difference. It is common to find the skewness and kurtosis values to vary widely especially for small number of cases (< 50), and is not a foolproof way to deciding on non-normality.
Table 4-25 Descriptive Statistics for Number of Evaluative Comments and tests for assessing normality.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Statistic</th>
<th>Std. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>EvalComments Mean</td>
<td>3.1228</td>
<td>.18576</td>
</tr>
<tr>
<td>5% Trimmed Mean</td>
<td>3.0780</td>
<td></td>
</tr>
<tr>
<td>Median</td>
<td>3.0000</td>
<td></td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>1.40242</td>
<td></td>
</tr>
<tr>
<td>Skewness</td>
<td>.660</td>
<td>.316</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>1.666</td>
<td>.623</td>
</tr>
</tbody>
</table>

A one-way analysis of variance was conducted to determine the impact of advice structure on the number of evaluative comments made. Levene’s test for homogeneity of variances among the three conditions did not find any significant differences at p < .05 level.

ANOVA found a significant difference at p < .05 level among the three conditions for mean number of evaluative comments made, F (2, 54)= 6.87, p= .002. Tukey post-hoc test revealed that the participants using the claim condition made significantly more evaluative comments compared to both the pattern or control condition. The mean for the control condition was marginally greater than the pattern condition, but the mean differences for number of evaluative was not statistically significant. Table 4-26 shows the mean and standard deviation for number of evaluative comments made.
Pearson’s correlation was computed between number of evaluative comments made and rationale ratings to explore the relationship between the two and the other measures used.
for assessing “goodness” of the advice. The number of evaluative comments was found to be moderately correlated with rationale, and this was statistically significant at p < .05 level. However, it only explained 12.4% of the variance in rationale ratings indicating the strength of correlation was also moderate. Number of evaluative comments was also moderately correlated with both usefulness (R=16.08%) and quality (R=11.28%), which were statistically significant. Context ratings were weakly correlated with the evaluative comments, and not significant.

Table 4-27 Correlation Coefficients and Variance Explained between number of evaluative comments and measures for goodness of design advice.

<table>
<thead>
<tr>
<th>EvalComments_Refs</th>
<th>Context</th>
<th>Rationale</th>
<th>Usefulness</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pearson Correlation</td>
<td>.060</td>
<td>.353**</td>
<td>.401**</td>
<td>.336*</td>
</tr>
<tr>
<td>Sig. (2-tailed)</td>
<td>.659</td>
<td>.007</td>
<td>.002</td>
<td>.011</td>
</tr>
<tr>
<td>Variance Explained (R)</td>
<td>6%</td>
<td>12.4%</td>
<td>16.08%</td>
<td>11.28%</td>
</tr>
</tbody>
</table>

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).
### 4.6.5 Impact of Advice Structure on Design Issues or Implications Considered

A one way between groups analysis of variance was conducted to see whether the independent variable (advice structure) influenced how much participants discussed design implications, measured by number of unique issues identified (#issues). The independent variable had the following three levels: pattern, claim, and control.

Based on data, at least ten design issues were identified that were considered by the participants. These issues were visibility, organization, flexibility, complexity, efficiency, familiarity, error, constraints, aesthetics, and consistency. A description of these codes is presented in appendix L. These ten issues represented the range of unique issues either implicitly or explicitly addressed in the design advice. No one advice had all ten issues.

On an average, the participants discussed approximately 3 unique design issues. The underlying distribution for issues identified was slightly negatively skewed and flatter than a normal distribution. The 5% trimmed mean was similar to the actual mean. Based on this, the distribution was assumed normal for further analysis.

Table 4-28 Mean and Standard Deviation for number of issues identified; Testing for Normality using Skewness/Kurtosis and 5% trimmed means.
Levene’s test for equal variances (HOV) did not find a significant difference, and hence the data did not violate the assumption for ANOVA. ANOVA found the mean for \#issues to be significant at the p < .05 level, F (2, 54)= 6.32, p= .003 . The Tukey post-hoc test confirmed that \#issues for the claim condition was significantly higher than the pattern or control condition. The mean differences in \#issues between the pattern and control condition was not statistically significant.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Statistic</th>
<th>Std. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>#issues</td>
<td>Mean</td>
<td>3.6140</td>
</tr>
<tr>
<td></td>
<td>5% Trimmed Mean</td>
<td>3.6267</td>
</tr>
<tr>
<td></td>
<td>Median</td>
<td>4.0000</td>
</tr>
<tr>
<td></td>
<td>Std. Deviation</td>
<td>1.44858</td>
</tr>
<tr>
<td></td>
<td>Skewness</td>
<td>.020</td>
</tr>
<tr>
<td></td>
<td>Kurtosis</td>
<td>-1.051</td>
</tr>
</tbody>
</table>
To explore the relationship number of unique issues and usefulness ratings, Pearson’s correlation coefficient was computed between the two. The #issues was highly and positively correlated with usefulness, and this was statistically significant at p < .05 level. It explained almost 47.05% of the variance in usefulness ratings indicating that the
strength of correlation was high. The number of issues was also highly correlated with both rationale (R=31.24%) and quality (R=35.76%), which were statistically significant. Context ratings were weakly correlated with the #issues, and was not significant.

Table 4-30 Correlation between # of design issues discussed and usefulness

<table>
<thead>
<tr>
<th>#Issues</th>
<th>Context Pearson Correlation</th>
<th>.224</th>
<th>.559*</th>
<th>.686**</th>
<th>.598**</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sig. (2-tailed)</td>
<td>.093</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td></td>
<td>Variance Explained</td>
<td>5.01%</td>
<td>31.24%</td>
<td><strong>47.05%</strong></td>
<td>35.76%</td>
</tr>
</tbody>
</table>

**. Correlation is significant at the 0.01 level (2-tailed).

4.6.6 Summary of Content Analysis

The additional analysis attempted to reconcile the expert ratings with findings from a quantitative content analysis. The following list shows the correspondence between dependent measures in RQ 1 and measures from content analysis:

- Context of Use (Ratings) with Context_Count (No. of words coded describing context-of-use)
- Rationale with No. of Evaluative Comments
- Usefulness with No. of Unique Design Issues

Participants in the pattern and claim condition wrote more, or described more, than those in the control condition (i.e., word count). Pattern condition discussed context significantly more than the control condition. They also seemed to discuss context more than claim, but the difference, measured by context_count, was not significant. Participants in the claim condition made
significantly more number of evaluative comments and identified more number of unique design
issues than either the pattern or control condition.

Comparing the measures from quantitative analysis with measures from content analysis
revealed the following, also summarized in Table 4-31:

- Word count was highly correlated with the ratings received on rationale, usefulness and
  overall quality.
- Number of words that were attributed to describing context was highly correlated with
  the context ratings.
- Number of evaluative comments was only moderately correlated with rationale ratings.
- And finally, number of unique design issues discussed was highly correlated with ratings
  on usefulness.

Table 4-31 Summarizing the relationships between the dependent measures in RQ1 and content
analysis

<table>
<thead>
<tr>
<th>Content Analysis</th>
<th>Correlations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Context</td>
</tr>
<tr>
<td>Word Count</td>
<td>High*</td>
</tr>
<tr>
<td>Context_Count</td>
<td>High*</td>
</tr>
<tr>
<td>No. of Evaluative Comments</td>
<td>Low</td>
</tr>
<tr>
<td>No. of Unique Design Issues</td>
<td>Low</td>
</tr>
</tbody>
</table>

* significant at p < .05 level
CHAPTER 5: DISCUSSION OF FINDINGS

It’s been long argued that pattern, as a structure, may help capture and share HCI design knowledge. This assumption appears to be shared between research on patterns in HCI and claims reuse. It's also asserted that patterns may be preferred over guidelines or design principles for communicating design advice (Pemberton, Griffiths, van Welie, Mahemoff). That is, presenting design advice in context along with supporting rationale may make patterns like advice more concrete and comprehensible to novice designers (Erikson?). Yet, almost a decade’s worth of discussion on this subject has yielded few controlled studies examining such promises (Dearden and Finlay, 2006).

This chapter discusses findings from a controlled experiment that evaluated the impact of a pattern structure on the communicating design advice. It explores a broader issue whether patterns add value when communicating of design advice. This study differs from past research on patterns in HCI by examining the benefits of using a pattern structure on communicating design advice, and not on the efficacy of available patterns. In the process, it addressed another criticism of patterns research that none of the prior studies have compared patterns with other forms of design advice in HCI (Dearden and Finlay, 2006). For this reason, patterns were compared with claims.

There are differences between patterns and claims, given their respective roots in architecture and design rationale. When capturing advice, patterns emphasize a problem-solution pairing in a certain context whereas claims focus on capturing the positive and negative implications to a design decision. But despite the differences, there also seem to be interesting similarities between the two, the very same properties that set them apart from guidelines or design principles:
• Discuss trade-offs among design issues or implications
• Provide design advice in context (when) along with rationale (why)
• Argue for abstracting design knowledge from successful HCI designs

The choice of claims was also motivated by the comparable complexity in structure between the two approaches, which ensured that minimal to no training was required by the participants for the claims condition.

The objective behind this research was to investigate the value of using a pattern structure for communicating interaction design advice. This was operationalized in the following two research questions that examined:

3. How does the advice structure affect the effectiveness of design advice?
   a. To what extent does the advice structure impact the context of use described?
   b. To what extent does the advice structure impact the rationale expressed?
   c. To what extent does the advice structure impact the usefulness of advice?
   d. To what extent does the advice structure impact the overall quality?

4. Does the advice structure affect the time required to prepare design advice?

In the following sections, we provide a summary of key findings from the study as it pertains to usefulness of advice, rationale expressed, context of use captured and overall quality. Based on the findings, we expand the discussion in following ways; all argued under the theme of communicating design guidance or rather how we can potentially communicate design guidance better:

• Exploring the contribution of cons to design advice
• Building a new design vocabulary which is based on representing design trade-offs in context
• Bridging between theory and a concrete design instance, and the role of problem framing when abstracting design guidance

We end the discussion with a proposal for an enhanced structure for capturing and communicating design advice, one that acknowledges cons and problem as important contributors to the usefulness and usability of design advice. In doing so, we further clarify how design trade-offs represent design advice, which can exist midway between concrete design instances and design methods.

5.1 Summary of findings

Inter-rater reliability, measured by intraclass coefficient, between the two expert judges was 0.81 overall, which showed a good level of consistency.

Of all the dependent measures (context, rationale, usefulness, overall quality, and time), analysis of variance (ANOVA) of the rating data found the differences in usefulness to be significant, $F(2, 18) = 6.783, p < 0.0125$. The $p$ value was set at .0125 level as part of Boneferroni’s correction (Fields, 2005) to account for type-II error. A post-hoc analysis using Scheffe’s post-hoc criterion showed that the advice prepared using the claim structure received higher mean ratings than either the patterns or control groups. Table 2 below shows the $F$ and $p$ values from ANOVA for the other variables that were determined as not significant. The advice structure did not affect the time required to prepare recommendation. The overall quality between the three was also not significant.

Table 5-1 Results from ANOVA for each dependent variable

<table>
<thead>
<tr>
<th>Measure</th>
<th>F Value</th>
<th>p Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 5-2 Descriptive statistics showing means (M) and std. deviations (SD). A 7-pt Lickert type scale was used. Values in **BOLD** imply that is the highest value in that row.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Pattern</th>
<th>Claim</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context</td>
<td>3.00, 1.07</td>
<td>2.71, 1.13</td>
<td>2.68, 0.96</td>
</tr>
<tr>
<td>Rationale</td>
<td>3.50, 0.88</td>
<td><strong>3.84</strong>, 1.29</td>
<td>3.39, 1.08</td>
</tr>
<tr>
<td><strong>Usefulness</strong></td>
<td>3.13, 0.99</td>
<td><strong>3.95</strong>, 1.12</td>
<td>2.79, 0.85</td>
</tr>
<tr>
<td>Quality</td>
<td>3.18, 0.87</td>
<td><strong>3.34</strong>, 0.90</td>
<td>3.00, 0.83</td>
</tr>
<tr>
<td>Time (mins)</td>
<td>12.73, 5.14</td>
<td><strong>13.15</strong>, 5.07</td>
<td>11.14, 5.63</td>
</tr>
</tbody>
</table>

* Significant at p<0.05

5.1.1 Usefulness of Advice

The advice structure had an impact on the usefulness of design advice. Claim structure performed significantly better than the pattern or control. Advice was rated more useful based on how many relevant implications were considered. To distinguish usefulness from rationale, rationale
assessed whether reasons were given for picking a particular menu (evaluative comments) whereas usefulness evaluation was based on the content and relevance of the cited implications. Looking closer, compared to the pattern and control condition, claim advice also contained a significantly higher number of unique design implications, $F(2, 54) = 6.32, p = 0.003$. When comparing amongst the three experimental conditions, pattern structure asked participants to discuss “what the user was trying to do”. Pattern condition also asked participants to describe why the suggested menu was appropriate in the given situation. The claim structure asked participants to consider the advantages and disadvantages of their design choice. Participants in the control condition did not specify any set structure, and were simply asked to explain why they would recommend a menu option.

The findings indicate that claim advice benefitted from a discussion of disadvantages. As part of a content analysis of prepared advice, the evaluative comments touched on ten types of design implications that were considered by the participants (e.g., visibility, efficiency). These can be argued as the set of design issues considered by the participants as a whole. This categorization was used in determining that the number of unique design issues had a significant and strong correlation with usefulness ($r = 0.686$), rationale ($r = 0.598$), and overall quality ($r = 0.598$) ratings, at $p < 0.05$ level, which seem to corroborate what we discovered from expert judging. Context ratings were weakly correlated with number of unique design issues ($r = 0.224$) and were not significant either.

Table 5-3 Summarizing the relationships between the dependent measures in RQ1 and Content Analysis

<table>
<thead>
<tr>
<th>Additional Analysis</th>
<th>Correlations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context</td>
<td>Rationale</td>
</tr>
</tbody>
</table>

### 5.1.2 Rationale Presented in Design Advice

Expert ratings did not reveal a significant difference between the three conditions when expressing rationale. In other words, the structure did not impact how much rationale was provided. In examining how participants argued for their design decision, the claim condition did elicit significantly higher number of evaluative comments (positive + negative) compared to the control condition or pattern condition, $F (2, 54) = 6.37, p = 0.002$. An example positive evaluative comment was stated as, “This design choice will allow the user to know all of their options at any one point in time (e.g., multiple categories can be open at once). The user can scroll through the entire open category at once.” The pattern and control condition almost rarely discussed negative implications of design choice. The difference in the number of evaluative negative comments was significant, $F (2, 54) = 114.882, p= 0.001$.

We observed that the claim structure prompted participants to be more reflective about their recommendation, both resulting in a higher number of evaluative comments, and as we saw earlier, lead to consideration of more design implications. This leads us to theorize that it may help to make explicit negative implications of design decisions in design advice.

### 5.1.3 Context of Use in Advice
The difference in the mean ratings for context of use based on advice structure was not significant. And while the context-of-use ratings were found to be higher for the pattern condition, this margin was not significant. Examining the content of design advice showed that participants in both pattern and claim condition discussed their design more than the control group (c.f. table 4 on word-count). The difference in mean word count was significant between patterns-control and claims-control, F (2, 54) = 4.27, p = 0.02.

Table 5-4 Total number of words present in design advice from three experiment conditions

<table>
<thead>
<tr>
<th>Condition</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern</td>
<td>200.3684</td>
<td>78.09838</td>
<td>19</td>
</tr>
<tr>
<td>Claims</td>
<td>200.4211</td>
<td>72.79752</td>
<td>19</td>
</tr>
<tr>
<td>Control</td>
<td>143.1579</td>
<td>56.26649</td>
<td>19</td>
</tr>
<tr>
<td>Total</td>
<td>181.3158</td>
<td>73.63791</td>
<td>57</td>
</tr>
</tbody>
</table>

The content of advice when compared on aspects that may be commonly associated with context (e.g., user, task) showed that pattern advice discussed context related aspects more than the control group. The number of words describing context also showed a strong correlation with context-of-use expert ratings (r = .509, p = .001).

5.2 Communicating design advice

Based on the findings, claim-advice was rated higher for usefulness, included more number of evaluative comments and in the process identified more number of unique design issues. As we have shown, number of unique design issues addressed by the participants was highly and positively correlated with the usefulness ratings. The claims group ended up providing more number of evaluative comments than the pattern or control group. Only participants in the claims group made negative evaluative comments, or in other words, included a description of cons. This fact in itself is not startling since the claim structure explicitly asked for cons. What is interesting
is that none of the participants from the pattern or control groups mention cons when arguing for their design recommendation. We theorize that an opportunity to discuss cons gave the claims group an edge, as we discussed in the beginning.

Table 5-5 Correlation between # of design issues discussed and usefulness

<table>
<thead>
<tr>
<th>#Issues</th>
<th>Context</th>
<th>Rationale</th>
<th>Usefulness</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pearson Correlation</td>
<td>.224</td>
<td>.559∗∗</td>
<td>.686∗∗</td>
<td>.598</td>
</tr>
<tr>
<td>Sig. (2-tailed)</td>
<td>.093</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>Variance Explained</td>
<td>5.01%</td>
<td>31.24%</td>
<td>47.05%</td>
<td>35.76%</td>
</tr>
</tbody>
</table>

∗∗. Correlation is significant at the 0.01 level (2-tailed).

In the following sections on “communicating design advice” we make a case for the importance of cons as an important contributor in understanding and arguing for design recommendations. Alongside, we re-visit and explore how and why advice should represent trade-offs, and propose a structure that can be used to capture this advice.

5.2.1 Cons in Design Advice

This study indicated that advice that discussed cons, explicitly, was perceived to be more useful than ones that did not. This section explores the possible role of cons in communicating design advice. To be specific, it discusses the following four ways in which we theorize cons participate in design advice:

- Discussing cons provide greater opportunities to identify implications
- Cons as a strategy to scope context of use
- Cons (or lack of) as indicators of maturity of design (advice)
- Cons aiding re-design and re-use efforts
5.2.1.1 Opportunities to Discover Design Implications

Given the freedom, participants displayed a tendency to “sell” their idea, and in the process were less critical. It appears that the pattern structure, as presented, did not offer a strategy for participants to easily identify and explore possible design implications. The control condition also displayed a similar positive-implications-only tendency. Since the control group was not prescribed an explicit structure, we can theorize that offering positive evidence alone may be a “natural” strategy when arguing for design recommendations.

The findings, however, show that “natural” may be not be enough. We found that a pros and cons approach in claim structure seems to have prompted participants implicitly guided participants to consider more number of unique design implications. In other words, making their recommendations more balanced and rounded. And while a larger number of design implications may not bolster usefulness, there was a significant correlation between the number of unique implications and rating of usefulness. We feel that adopting a pros and cons approach to describing advice may provide greater opportunities to be reflective and critical about design advice.

5.2.1.2 Specifying and Scoping Context of Use

The pros and cons format for implications may help us address the challenge of discussing context in a way that balances specificity of advice with generalizability. An exhaustive specification of context may lend concreteness to design advice, but it may also impede a designers’ ability to critique this advice in reuse situations. Furthermore, an exhaustive specification is impossible.
We found that a pros and cons approach prompted participants to implicitly describe leading aspects of context. For example in stating a positive implication, the participant states, “the +/- button on a directory-tree menu will be familiar to users' based on their prior experience to indicate more or less items in an organized list of items.” Here, in offering a positive evaluative comment, the participant touched upon aspects of context like users’ experience level.

When we consult pattern collections, a commonly found structure is that of a problem, solution, and context. On the surface, this mirrors Alexander’s stated definition of a pattern. Pattern collections have gone further to specify a “use-when” or “why” section which attempts to outline the context when or where the advice is relevant. However, a re-reading of Alexander’s definition seems to hint at a something different, which is, a pattern in essence is a problem and solution discussed in context, not and context. The difference between these two readings of the definition may seem subtle, but there is a cost. By adopting and-context perspective we may be suggesting that problem and solution can potentially be abstracted from context, and this separation is desired. We, however, feel that this separation is artificial and difficult to achieve.

The pro and cons format for discussing parts of the advice, in the study, show that this separation between context and problem/solution is unnecessary. The design advice prepared using this structure showed that it was more natural to pair each design implication with the context of use. In fact, the design issue and implication lost its meaning without specifying the context of use alongside; in other words, the argumentation loses its purpose. The “goodness” or benefits of the design feature (i.e., +/- or up/down arrows) is only relevant when information is organized as a list with category names. Without discussing this aspect of the context, there is little hope in arguing why a particular design feature is necessary. We found that discussing pros and cons allowed participants to capture relevant aspects of the context without explicitly aiming for it. We theorize that stating pros and cons helps to keep the design issue embedded in the
context without special effort. This relationship between design advice and problem-solution in context will be discussed in more detail in section “5.2.2. Design Advice as Shared Vocabulary.” The discussions on patterns in HCI thus far have eluded providing any concrete recommendations on how to discuss and scope context.

One the one hand, we found that drawing attention of our participants to explicitly account for context in the pattern structure did not make a significant difference. On the other hand, the claim structure seems to have achieved what a pattern-advice was meant to be, describing advice in context. Furthermore, pattern-advice and claim-advice did not differ much when discussing advice, in terms of length of explanation (approx. 200 words). However, as described in the earlier section, including cons had an additional benefit—advice perceived as more useful. In summary, pros and cons appear to be an easy mechanism to scope the unbounded nature of context. The advice may only need to capture the essential bits of context that rub against the problem and solution being described.

5.2.1.3 Assessing Maturity of Advice

In an earlier section we theorized that including cons is a good thing. In this section we venture two situations where discussing cons may be redundant. In other words, if the design situation does not meet these two conditions, we should account for the cons explicitly:

- We are implicitly aware of the cons or residual forces through lived-in experiences
- Advice communicates a perfect solution that helps bring the design forces into balance; such the net forces are zero, (i.e., no negative implications).
To explain the first case, where we are implicitly aware of the cons, consider the following example: Imagine walking up to a home through the main or front entrance and entering directly into a bedroom, as shown in the figure above. Will we be surprised to find such a house? Will we be able to enumerate the possible cons of this decision? In the cultural context we live in, it may be uncommon (not impossible) to find such an example, and as such may not meet the common expectation. If we have to phrase a question, when providing advice that suggests placing the bedroom away from the main entrance (in a more private space), do we need to be made explicitly aware of cons? We would feel that in most cases, no. In other words, we have stopped thinking about such cons when designing a home since we have certain expectations about how the private rooms are planned or located. We argue that in such situations, it may be OK to leave out an explicit mention of cons because it might be redundant to the reader.
In contrast to architecture, our ability to discern and articulate the cons may not come as naturally in interactive systems. Mainly because the number and variety of experiences we have with interactive systems pales in comparison to that of architecture. In other words, we may not have the benefit of lived-in experience (usage), and hence, may be unaware of the negative implications.

Consider the example in figure X for an interactive system. Can we confidently discuss the pros and cons of this approach as easily as in the architecture example? It is not surprising if the answer is “no.” It may even be understandable if we chose to seek more information in this case before articulating cons. For instance, what is this an interface for? Who is going to be using this? How many products have to be shown on this interface? In the architecture example, we may even be relatively less reflective, but still be capable of filling in bits and pieces of the context based on our experience in order to list the cons. If the answer to the earlier question is yes, it may be indicative that we have had sufficient experience using such controls in varying contexts, enough to understand the benefits and limitations.

Consequently, we can argue that the ability to discuss the cons can be used as an indicator of the maturity of design advice. It will be difficult to offer advice as good advice if it
has never been put in use. And without having implemented this advice, any discussion of potential cons may be conjecture. With the exception of Jeffries (2004), we haven’t come across literature that discussed the value of cons when giving design advice, which includes research reported in CUE studies, which focused specifically on giving design recommendations. Even claims do not go into the details of the benefits of including cons in communicating design advice. To summarize, we feel that the presence of cons is simply evidence of use. The added benefit is that we can be relatively more confident using advice that includes cons compared to that which does not. Furthermore, it’s mature advice when we are able to articulate cons just as easily we are able to do so in architecture, or at least with similar ease.

5.2.1.4 Seeding Re-Design

Given that there is no perfect solution, cons may be understood as the un-resolved forces post trade-off. And addressing cons can be argued as one goal of an iterative design process.

The design problem (p0), as we see it, is the trade-off between forces that tend to pull the solution (s0) in opposite directions. In a desktop publishing example, as system functionality (i.e., menus and commands) is made more visible on the interface, it may start to appear all the more complex to a first time user. And assuming that for the given context of use (i.e., typing a document) achieving simplicity was more important than initial visibility of functionality, the cons emerge from the lack of visibility. In other words, for every solution (s0) we implement, we create new forces that now need to be resolved. Balancing these new forces is the next design problem (p1), which may inherit constraints (pros) from the initial solution (s0) and so on.
In this sense, a discussion of cons serves two purposes. First, it describes the known limitations of the suggested solution. And secondly, it potentially serves as a natural starting point for improving this solution. In other words, the re-design effort is directed at addressing the cons in the next iteration of the solution without sacrificing the pros of the current solution.
Returning to the desktop publishing example, having a toolbar with "frequently used" commands is one solution (s0) that balances complexity and visibility of system functionality. Some of the cons (p1) are the following:

- Less frequently used options are hidden; commonly used commands may be a small subset of the available common. Consequently, users may be unaware of the additional commands. Using these special commands may allow the task to be more efficient.
- We need an organizing principle for presenting/representing the commonly use commands.

Iterating on this, the solution (s1) to this new problem (p1) may be to introduce the concept of "in-context menus" that surface when needed, and grouping commands in tabs (e.g., common, insert, charts). This is common practice for surfacing commands. And now if we are to improve this solution, an easy approach may be to address cons identified in use from using s1.
Thus, moving along the iterative design cycle; on that may be powered by cons. In an evaluation-driven practice like interaction design, it can be argued that the goal of evaluation is not only to assess the solution fit, but also to determine misfits (usability problems). In this sense, cons are a natural part of design and should be included.

5.2.2 Trade-Offs in Design as a Shared Vocabulary

We argue that HCI as a community may have a shared vocabulary to discuss issues impacting usability in interactive systems (e.g., design principles). However, we do not have an explicit vocabulary to discuss design trade-offs. Trade-offs, we assert, reflect both design as a noun (designed form) and verb (designing) more closely than the design principles themselves. And consequently, trade-offs need to be at the root of communicating design advice. Furthermore, these trade-offs need to be situated in a context of use to make it meaningful. Achieving this in design advice may address two common challenges faced by designers—deciding on and applying design principles, and resolving conflicting design principles. The following discussion focuses attention on the managing trade-offs amongst multiple design principles.

5.2.2.1 Interpreting Design Principle(s)

Providing affordances, appropriate mapping or visibility of system are some examples of design principles, which have been found to impact usability of a an interactive system. For purposes of explanation let us consider the "visibility" principle, which in essence conveys that an interactive system that has its primary functionality visible (i.e, apparent or easily noticed) to the user may be perceived to be relatively more usable than a system which does not. In order to understand the visibility principle more concretely, consider the MS Word application. If all the toolbars and
icons in MS Word were hidden, it can make discovering how they can change the font size or color of the text more difficult. Based on this principle alone, System shown in figure A can be argued be more usable than the interface shown in figure B for the purposes of typing a text document. And at some level, these principles are offering a form of design guidance, guiding the designer towards designing systems that instantiate these stated principles.

Example A

Example B

Figure 5–6 Example A shows a traditional MS Word 2007 with the Ribbon interface. Example B shows a simple text editor in OS X, similar to the Wordpad application in the Windows OS

However, the visibility principle in itself does not mention anything about how this principle is applicable for text-processing application context or rather does not restrict it to this particular context. In a way, demonstrates the power of guidelines, that is, the ability to generalize it across a wide range of interactive systems. But generalizability comes at price as it lends less guidance in implementing a design principle. To a large extent, guidelines tend to be domain independent, which is in contrast to design that by nature tends to be domain/context sensitive.

With guidelines, it’s the responsibility and skill of the designer to implement (satisfying) the relevant principles to fit the context of use. Despite the level of experience a designer may have in
the past with principles like visibility, in practice we are quick to find that there are no obvious right or wrong ways for realizing a principle.

Let's consider the design of Microsoft Word’s writing interface. We can argue that displaying all the functions available may satisfy the visibility principle, which may suggest that there will be no more hidden menus to deal with. While it may seem reasonably clear to the designer that making functions visible is good a thing from an interaction perspective, it is not apparent how visible the functionality needs to be.

![Figure 5–7 Example A is shows a common instance of MS Word; Example B shows the MS Word application with all the toolbars toggled visible.](image)

In the two examples shown above (refer Fig Xa and Xb), is there a difference in how the visibility principle has been realized? One obvious answer is that in Fig Xb almost all of the functionality
is revealed on the interface, and in the other example only some of it are shown. Consequently, we could argue that on one functionality is more visible vs. less visible on the other. For the sake argument, let’s assign values of 0 and 1 as the limits for visibility principle. On this scale, a value of 0 will imply that all the functionality is completely hidden from the user, and a value of 1 will imply that the functionality is perfectly visible. Using this scale, we can theorize that example A lies somewhere closer to value 1 and example B may be closer to 0. The problem for HCI practitioners is that both design instances that lie at the extremes (0 and 1) are sort of theoretical. It may be difficult to draw, articulate or defend what such an interface should look like. In practice, this scale (e.g., visibility scale or complexity scale) is a continuous one, where a wide range of solution instances is possible depending on how designer chose to implement this principle. The problem seems to be that the design principle in itself does not make this obvious. In realistic design situations, principles have to be interpreted in shades of gray. And consequently, we argue that design advice should guide designers towards tangible solutions. Without that, principles tend to represent descriptions of common problems that we face in interactive systems or issues designers need to consider during design.

5.2.2.2 Applying Design Principle(s) in Practice

The prior discussion about applying a single design principle (i.e., visibility) sets the stage for the issue at the heart of communicating design advice--about trade-offs. That is, principles by themselves do not offer a way for designers to negotiate trade-offs in design. In other words, when guidelines conflict, and they most often do. Textbook examples of design problems, however, often seem constructed so that novice designers may understand how each design issue relates to ease-of-use. Such problems tend to be just complex enough such that addressing one particular guideline appears to resolve the problem at hand. While this approach is fine for
introducing novices to HCI or design principles, as a first step, but may still leave them ill-equipped to tackle more realistic design problems. And while have seen that following design principles (e.g., affordance) or guidelines impact usability design, design problems are quick to show that the issues we have been treating or learning about independent of each other in practice affect each other. That is, favoring one design principle will impact another and vice-versa, and there in lies the real problem. And we need a way for capturing, communicating, and educating the practitioner when we talk about design advice.

![Diagram of complexity and visibility principles](image)

**Figure 5–8** Illustrating the relationship between complexity and visibility principles in design practice. Value of 0 indicates a design were this principle has not been considered, and a value of 1 indicates when a principles has been perfectly realized.

We assert that mere the knowledge of design principles alone does not provide guidance for resolving common conflict situations in design practice. As a thought exercise, consider the situation illustrated in figure X. In this example, we will discuss two principles that seem to pull the solution in opposite directions, visibility and complexity. For concreteness, we can think about the text-editing interface example that was described in the prior section. Let's first qualify this relationship between visibility and complexity. Which is, a system that has all its functionality visible may make the system appear more complex and vice-versa. Ideally, we want
a system that scores high on visibility, but low on complexity. Using our reasoning from the previous section, now we have two principles each with its own theoretical limits of 0 (completely absent) and 1 (fully present). The task for the designer is to maximize visibility while minimizing complexity. Although terms used here suggest an optimization problem, it's a satisficing problem. Drawing this distinction is not the focus here.

The complexity-visibility balance is an example of how design principles may compete with each other. It’s no different from an automotive engineering problem where the goal seems to be building engines that are fuel-efficient but also have adequate power. In other words, the trade-offs seem to be between efficiency and power. In our experience, these are competing goals and achieving fuel efficiency without sacrificing power IS the design challenge. In the same vein, the design problem for interactive systems is not realizing the visibility OR complexity principle, but rather, resolving visibility AND complexity at the same time. We have already argued in an earlier section how realizing a single principle is not as straightforward. But now we have at least two principles that are in play simultaneously and pulling the solution in opposite directions. So, how to balance these two principles/forces? We assert that design advice or guidance needs to be about managing trade-offs.

5.2.2.3 Capturing Trade-offs AS Design Advice

The previous section presented a simplified description of how consideration of design principles surface in design practice. It is one that involves achieving a trade-off between at least a pair of forces that often tend to pull a design solution in opposing directions. The final solution is system where these opposing forces are in balance (satisficed). These trade-offs are always dependent on the context in which these forces exist. Discussing trade-offs independent of context loses
meaning very quickly. In this section we will review the how the pattern or claim structure
highlight this in their own way when communicating advice. We believe that capturing advice as
trade-offs in context can compliment existing practices (i.e., principles) for sharing our
knowledge in HCI. Furthermore, we will discuss how an artifact-driven approach to capturing
design advice fills a gap between concrete instances of design and abstract principles, which is
shareable and tangible. It also seems to address some of the challenges described in the earlier
section.

In a manner of speaking, all design decisions are arguably correct with varying degrees of
correctness depending on the context. While the arguments supporting patterns or similar formats
have appreciated the value of context in design advice, not enough attention has been given to
benefits of articulating trade-offs to communication, both for shared problem understanding as
well as sharing the solution. Furthermore, we feel that it’s difficult to understand trade-offs
without talking about context (i.e., use when). Mainly because there is little chance of confirming
whether a design decision is good without evaluating its impact on the intended context of use. In
the Microsoft Word example, where the designer has to trade-off between complexity and
visibility, the context is set by the purpose or activity supported by the application, which is
typing a document. So the design problem can be interpreted as to maximize the viewable area
for writing while revealing the most commonly used features when typing a document, and
hiding the rest. Without discussing the writing task (context), the problem space is unbounded
(i.e., level of visibility or level of complexity). Both patterns and claims are potentially viable
approaches for communicating design knowledge; not only because they tend to include more
context than guidelines, but also because they tend to address trade-offs in context. This
contribution appears to be under-reported and implicit in related literature in terms of value to
communicating design advice. Given that design problems are messy (a set of intersecting design issues), good design can be argued as making the right trade-offs.

Both pattern and claim embody trade-offs, but each in their own way. Claim explicitly communicates the implications of the solution by discussing the pros and cons. Patterns traditionally mention the trade-offs upfront in the problem description, and then provide rationale primarily to provide evidence for the solution. The solution in a pattern describes the essence of successful trade-offs in similar contexts in practice. In other words, in a claim-advice the trade-offs are about the solution and in a pattern the trade-off is the problem. This indicates two complimentary approaches to providing design advice. The general process for capturing this guidance is at least a two-step process: First, to identify good solutions; and second, to capture it in a format that is shareable and re-usable like guidelines. This study attempted to understand this contribution of structure (pattern and claim) to communicating advice.

5.2.2.4 Bridging Existing Artifacts and Theory in Design Practice

One approach for discovering trade-offs is by extracting it from design artifacts in use. Designed artifacts, we argue, represent design solutions. In other words, artifacts we see around us, virtual or physical, represent solutions where specific trade-offs have already been made. In practice, designers use both principles and concrete designed instances to inform their decisions. The principles provide the base-line vocabulary to discuss usability issues, and the solution instances represent trade-offs in context. Let us explore this process briefly.
Figure 5–9 A graphical representation of Norman’s Gulfs of Execution and Evaluation and the various stages that help bridge these gulf.

Figure 5–10 Using design instances or examples of clear and “good” navigation being used on different websites from http://patterntap.com/tap/collection/navigation
Consider a scenario where a designer is designing a navigation for a corporate website. The designer could keep the principles in mind (Fig 5-9) for designing a usable navigation or consult at examples of seemingly usable navigation (Fig 5-10) on other corporate websites. Using the first approach, the designer has to determine which principles are going to be relevant, overcome the challenge of contextualizing each relevant principle for the task at hand, and manage the trade-offs between the principles. Using the second approach, the designer compares across the various examples, pulling out bits and pieces of what she believes works and recombines them in a meaningful way to suit her context of design. From the pieces abstracted from a unique instance (example), the designer strips away aspects not relevant for the given design space or retain aspects of it depending on fit between the source and target context of use. This we believe is a critical step, where the designer forms the "blocks" from design instances (source), which then are contextualized and used in the design solution being crafted (target). These "blocks," we believe, represent pieces of guidance that represent trade-offs in context. These "blocks" emerge as a new vocabulary for describing and rationalizing design. Figure X shows an illustration of the process.
Figure 5–11 Each node represents a design principle, and each block represents a group of principles that have relationships defined that have been extracted from the solution instances. The size of the nodes indicates the priority assigned during trade-off.

We believe that in reality designers are never really using a single approach (theory-driven advice OR artifact-driven advice), but blending the two. The principles provide the vocabulary used to frame the design space, thus identifying the relevant usability issues, which in turn guide selection of the relevant “blocks” abstracted from design examples/instances. Concrete artifacts, like Microsoft Word, thus provide us an opportunity to learn about design from design (artifact). It affords us unique opportunities to learn about use and experience an artifact situated in a specific context. Trying to learn from designed artifacts appears to offer following four benefits to comprehension: form, function, context-of-use, and the inter-relationships. MS Word, while being a unique design instance, has a clear and visual description of the various features that users can interact with, has a stated purpose or function that situates the various features, and analyzing
the type of document produced helps us construct a rough idea about the context-of-use. And while we may discuss each of these independent of each other, the final artifact always provides a coherent view of the relationships among attributes, functions and context-of-use. This coherence may prove valuable when attempting to communicate design knowledge. It’s an opportunity to learn from the trade-offs made. Carroll & Kellogg (1996) argued for a similar middle ground when they proposed the concept of “claims.” They assert that claims could potentially serve as an artifact that could exist between these two extremes, theory-driven vs. hermeneutics-driven approaches. Our argument while similar to this one, seeks to broaden it by calling this middle ground design advice.

Both patterns and claims advice seem to exist in this space between principles and concrete artifacts, and represent this "new" vocabulary. Both offer a traceable connection between concrete artifacts and design principles. This was the justification for comparing pattern structure and claim structure in this study. We found that design advice contains cons is perceived to be more useful to designers. And in an earlier section we discussed why cons are potentially helpful for an iterative design process. But before we assume that claim structure is complete enough to represent this a artifact-driven design advice, let's briefly discuss one of the key differences between the two structures. Claim structure seems to emphasize the pros and cons when communicating advice. The pattern structure on the other hand seemed to emphasize the problem being solved, and then rationalizing the solution (i.e., pros). In other words, both the pattern structure and claim structure attempt to convey similar content with regards to the trade-offs. However, pattern structure seems to require articulating the trade-off that needs to be made (i.e., problem), and claim structure is more robust in presenting the trade-offs that have been made (i.e.,
solution. And if we are to pursue this artifact-driven guidance, we assert that both instances of trade-offs need to be captured; both before and after trade-offs are made.

Figure 5–12 Design advice occupying a mid-level between principles and concrete design instances; the levels from bottom to top correspond to an increasing level of specificity of context. The icons on the right correspond to shareable (vocabulary), tangible (visual) and reusable advice.

5.2.4 The Role of Shared Problem & Artifact-Driven Advice

In an artifact-driven approach, the solution is obvious. It's the artifact itself. What is not obvious is the problem it solves. And design knowledge, we assert, lies in communicating the problem solved by the solution as much as the solution itself. Because, every solution is only as good as the problem it solves. This may raise a question whether discussing the problem is "required" whenever we discuss a solution. May be there exist situations where we all have an implicit
understanding of the problem being solved, causing the problem description to be redundant. This to some degree is true, and we are able to qualify this using a thought exercise from architecture.

In this exercise, we need to design the layout of a two-bedroom American household. A drawing representing a square plot of land bounded by a main road and an access road on opposing sides is the available space. Circular paper cutouts (c.f., Fig 5-13) representing the different living spaces (e.g., living room, family room) are also given. As part of the task, you have to arrange these cutouts to represent the layout of the house (Fig 5-14). Assuming that different individuals from a similar culture complete this task, we theorize that while there may be minor variations, given the orientation of the plot of land, we will end up with final layouts that are unremarkably similar. Moreover, the relationships between rooms (adjacency between specific rooms) will be preserved. Lived-in experience prompts us to layout the rooms that show a gradual transition from public (e.g., living room) to private spaces (e.g., family room) as one proceeds in from the front towards the back. Our rationale may also be unremarkably similar. Broadly speaking, the implicit design problem was to arrange the different rooms in manner that it's internally coherent (room to room) and at the same time externally coherent (room to outside world). What we understood from this exercise is that this problem did not need to communicate explicitly. There is an implicit and shared understanding of the problem. However, this is not the case always. That is, different individuals may see the design problem quite differently, and that may influence the solution they come up with. This should remind us of the "three blind men and an elephant" situation. To each blind person, the elephant represents a completely different thing. We argue
that such may be the case with problems in design.

Figure 5–13 Illustrating the home layout design problem.

Figure 5–14 A candidate solution for the given home-design problem.
Depending on how the problem is framed, it may influence the solution(s) we come up with. The example exercise of planning a home layout is indicative of this in a way. Different individuals may arrive at similar solutions provided there is a shared understanding of the problem. In the case of communicating design advice, the problem framing refers to the trade-off that needs to be made amongst the usability issues identified or desired outcomes in a specific context. An alternative approach is to claim that when the design problem is framed differently, it may lead to different solutions. We believe this was the case for the experiment presented in our study. Our data indicates that given the room planner task, there were no clear favorites among the three design alternatives. There were no clear choices. We can also claim that the design task was loosely defined enough around the task the user had to do, and did not prompt the participants to consider one menu option over another. To explore whether problem-framing influenced the selection of design alternatives, advice generated by the participants were analyzed in the following manner- we used Nielsen's heuristics to classify statements presented as part of the advice; for each menu option picked, we calculated the occurrence of each heuristic.

We found the final design choice in the study seemed to depend on which design issues were prioritized. In the study, each participant was offered three design choices: accordion, directory-tree and drop-down menu. The comparison showed that participants recommending a directory menu structure valued the ability to keep different product categories open when selecting furniture items. They also felt that being able to view related categories simultaneously would allow users to quickly make their selection. Participants advocating an accordion menu structure believed that being able to view the product categories at all times would be more helpful to the users. They also believed focusing users on one category at a time will keep the interface uncluttered and simple. The differences in how certain issues were prioritized give us an idea about how the participants internalized the design problem. The claim-advice structure,
however, prompted the participants to critique the solution and discuss the negative implications on use given the design choice. The question here is not whether these priorities were correct, but rather what were the priorities.

Making the priorities explicit sets the stage for readers of the design advice to debate its merits, and possibly reconcile differences. We argue that a problem framing exercise mediated the design solution that was finally recommended by the participants. As evident from their rationale (evaluative comments), it becomes clear that the design choice depended on design issues that each participant deemed relevant and important. That is, a participant who understood the design problem to be that of satisfying flexibility, efficiency, and familiarity design criteria, chose the directory-tree menu. We will point out that these criteria were determined by the participants, and not suggested in the task instructions. So, based on the task context (i.e., furniture shopping), participants interpreted the flexibility criteria as the ability to keep multiple product categories open in order to mix and match furniture items. Efficiency was suggested as being related to a flexible interface such that less time is wasted in opening and closing menus. And finally, familiarity was argued as users of the system having a history of experience with directory-tree menu structures in other interactive systems (e.g., Windows operating system). A similar argument can be also made for participants picking the accordion menu structure where the problem appears to be framed as satisfying visibility, complexity, and efficiency.

Both, in the case of pattern-advice and claim-advice, the problem is implicit in the description of the solution, and not usually discussed by itself. While pattern literature in HCI argues in favor of capturing the essence of the solution, they tend to lack a similar zeal when articulating the problem or the essence of the problem. Alexander’s proposal for patterns on the other hand pays special attention to framing the problem and setting the scope of the problem, both in terms of
context and specifying the conflict between leading forces. For example, in the ALCOVE pattern, the problem is described as a conflict between feelings of community, and the need for privacy in communal rooms. Both patterns in HCI and claim-advice seem to lack a description that distills the design problem down to its core structure. This core is an articulation of not only the forces/issues at play, but also how it impacts each other. We feel that the articulating the problem (essence) is a critical step in problem solving, and hence earns a place in communicating design advice.

Schön calls problem setting (scope + framing) an essential part of reflection in design. He succinctly defines problem setting as,

“… the process in which, interactively, we name things to which we will attend and frame the context in which we will attend to them.” (Schön, 1968)

He goes on to suggest that,

“In order to formulate a design problem to be solved, the designer must frame a problematic design situation, set its boundaries, select particular things and relations for attention, and impose on the situation a coherence that guides subsequent moves.” (Schön, 1968).

Given his views on the design problem solving, the design review session or reflective conversation explained in Educating a Reflective Practitioner between a master, Quist, and his student, Petra, can essentially be argued as an iterative process of framing and re-framing the problem to guide future actions (Schön, 1987, pp.44-79).

Schön is not alone in emphasizing the importance of problem representation. Both Simon and Norman emphasize the importance of problem representation as part of the solution discovery
process, if not the solution itself! The example of a game where two players alternately pick a number from 1-9 such that the first player to pick three numbers that add up to 15 wins the game can be called an exercise in problem representation (Simon 1996, Norman, 1988). Cross’s (2004) review of studies on expertise in design involving architects, engineers, and artists observed that experts pro-actively structure and formulate the problem.

![Diagram]

Figure 5–15 Comparing pattern-structure and claim-structure with aspects desired of design advice.

Moreover, considering situations where designers need guidance, it can be argued that the portion of design space known to them are the design issues that need reconciliation, and the context in which these design issues exist. Hence, making this explicit and part of design advice will potentially allow designers to assess the match between the problem at hand, and the problem being addressed in design advice. All this before they begin assessing the solution itself.
We find there is enough evidence to support the importance of problem framing, and feel this requires to part of design advice in order to situate the solution. Simply put, articulating the problem appears to a process of identifying and deciding upon a set of relevant design criteria that a solution needs to achieve, and then going on to scope the context within which these criteria make sense. At present, pattern or claim-advice in HCI do not devote much attention to setting up the problem, but focus more on describing the solution. The problem itself is either ill defined or implicit in the solution. This makes it unclear what exactly does this solution solve. The expectation seems to be that readers will be able to extricate it and piece together the problem without assistance. In summary, we have argued for following aspects that we expect of design advice:

- Identify the design issues at play
- Frame the problem (trade-offs in context). That is, specify how the design issues are related in this context.
- Discuss a good solution (trade-offs that were made)
- Rationale expressed as pros or advantages
- Rationale expressed as cons or disadvantages

5.2.5 Proposing an Enhanced Structure

The above discussion elaborated our hesitance in proposing claim-structure as the chosen format for communicating design advice. We argued there is more to communicating design advice than merely favoring cons when discussing the solution. This motivated the proposal for an enhanced structure that combines the problem in context with solution in context, along with rationale explaining the fits and misfits (as shown in figure 5-16). This section further explores this enhanced structure with concrete examples showing how the five aspects we desired of design
advice are realized. We illustrate this enhanced structure by using it to capture design advice for two interactive elements - accordion menu, and autocomplete text box.

![Diagram of enhanced structure]

Figure 5–16 Illustrating the enhanced structure, and how the independent pattern and claim structural elements participate in this proposed structure.

As a foundation for the proposed enhanced structure, we refer to advice structure used in the radiation problem (Dunker, 1945), which we believe to be quite similar. Gick & Holyoak (1980) showed how structuring advice in this manner, in addition to clarifying the advice, leverages the power of analogical reasoning, whereby we attempt a solution by abstracting structural similarities even between two or more semantically different problem domains. For readers unfamiliar with the radiation problem, it has been discussed below. It describes a problematic medical situation as follows (paraphrased):

- Problem Setting:
o Doctor treating a patient
  o Patient has tumor deep inside the body surrounded by healthy tissue
  o Doctor has access to radiation therapy

- Desired Goal: A tumor needs to be destroyed deep inside the body,

- Problem Essences
  o A high intensity radiation ray will kill the tumor but at the cost of healthy tissue it passes through.
  o A low intensity radiation ray is harmless to the surrounding healthy tissue, but does not affect the tumor either.
  o Radiation is the only option, and not to compromise healthy tissue.

Gick and Holyoak (1980) report that participants who received a story that contained an analogous problem setting and constraints, even when far removed from the medical domain (attack-dispersion story), succeeded in offering a feasible solution. The attack-dispersion story is as follows (paraphrased):

- Problem Setting:
  o Military general is organizing an attack on a fort
  o The general has a large army at his disposal
  o Fort is in the middle of an island surrounded on all sides by a forest
  o Access roads radiate from the fort through the forest like spokes of a wheel

- Desired Goal: Capture the fort deep within the forest with the army

- Problem Constraints (Essence)
  o Entire army cannot pass over a single road because large groups trigger land mines buried in them
  o A small group can pass over the roads without detonating the land mines.
The entire army force is needed to take over the fort. A small battalion will not work.

- **Solution**
  - Divide army into small battalions
  - Approach fort from all directions in small groups using the access roads

In both these situations, we can see there is value not only specifying the desired goal, but also understanding and articulating the problem, which discusses the trade-offs explicitly. Once we understand the problem constraints (criteria), the properties desired of the solution become more obvious. In this structure, the problem has been represented as problem constraints and the goal to be achieved given a particular setting. We will now discuss how we employ a similar structure to articulate advice for interaction design. For this we utilize following sections to emphasize different aspects of the advice:

1. Problem setting, provides a brief description of the situation/domain
2. Problem Essence, identifies the design issues at play
3. Design Goal, articulates/summarizes the design problem that highlights the relationships between the issues
4. Solution Scenario, explains how this is implemented or the trade-offs made with examples
5. Pros, fit between the solution and problem
6. Cons, misfits between the solution and problem

The following sections for ACCORDION MENU and AUTOFILL illustrate how this new formalism can be implemented. Appendix O shows a comparison of the structure that further
illustrates the difference between the three experimental conditions (i.e., pattern, claim, and control) more concretely

5.2.5.1 Accordion Menu Design Advice

Problem Setting
The interface consists of an empty staging or canvas like a room planner, where objects can be placed and manipulated. A collection of objects consisting of images and descriptions is available to choose from for adding to the stage area (e.g., symbols, shapes, or furniture items). These objects can be added or removed to and from this interface (canvas).

Problem Essence

- **Visibility**- Users needs to be able to see an overview of objects/shapes available to them. If absent, it forces the users to remember the options available to them. It is preferred that this collection of objects is conveniently located in one place instead of scattering them all over the interface.

- **Complexity**- when faced with a large number of objects, users may get over-whelmed with the variety or number. The challenge is to present these options in an organized manner such that users can easily switch between an overview and detail views.

- **Spatial Constraints**- On the interface, the staging area gets visual/spatial priority, and consequently, the amount of space available to show the collection of objects is limited. The collection of objects needs to be displayed without obscuring the main activity, for example, staging a room.

Desired Goal (Functionality)

Users need to be able to view a large collection of objects shown to them in a familiar and unobtrusive manner. They should be able to switch between overview and detail views quickly to
avoid being overwhelmed while making the best use of available visual space to display the collection of objects.

**Solution Scenario**

The user wants to construct a flowchart of a business process. The main drawing or charting area is empty. On the left side of the staging area, the user sees a vertical list of item groupings with each group visually separated from each other by a heading name. For an accordion menu (refer figure 5-17-1), one of these categories will remain in an expanded state revealing the object thumbnails within it.

![Diagram](image)

*Figure 5–17 Menus shown as part of 1) diagramming application and 2) Microsoft Word 2007 toolbar.*

Each group heading is visually differentiated from the items. A visual thumbnail of the items is shown instead of using text to describe the item. Once the users locate an object/shape required, they select and drag it onto the drawing area. Selecting or clicking on any other group headings
(i.e., ones not expanded) will collapse the previously open category and open the new selection.

At any one point, only contents from a single group are shown.

**Pros**

+ The group headings give a *visual cue* to the users about the type of objects available to them. The visual thumbnails of objects promote recognition instead of reading item names available.
+ Since only one category is available at a time, the user will not be overwhelmed with the number of choices, thus potentially reducing *complexity* of the interface. They can focus on one category at a time.
+ If an activity-centric grouping strategy is used, the user will not need to waste *time* switching between multiple categories. What they need to complete the activity is shown in one place while hiding items not relevant. Neither will they need to spend time to collapse a previously open category as clicking a new category automatically collapses the previous one.
+ In situations when the total menu height is fixed, the accordion structure balances visibility of group headings with details of the categories. The accordion makes best use of the available area minus the space used to display the group headings.

**Cons**

- The *organization* of objects or shapes needs to be evaluated to determine whether to group them by type or activity, which in turn depends on the task context. If it’s a diagramming application then the logical grouping may be based on type of object (e.g., callouts, basic shapes). Another possible grouping may be according to the activity (e.g., flow-chart, living room) such that shapes required to complete a certain activity are grouped together.
– Depending on the scope of the activity, the objects needed may belong to different categories (e.g., matching colors or style of furniture). This may require the user to keep more than one category open, which is not possible with the accordion menu structure.

– A large numbers of groups or headings (and menu height is fixed) may use up a significant amount of the vertical space. The remaining space may be too little to display the item details or to be useful to the users.

5.2.5.2 Auto-Fill Design Advice

Problem Setting

The user is entering/typing information into an input field of a text-based form, for example, filling out an email address or an URL in a browser address bar (refer figure 5-18). The form fields either expect text input with no typographical error or in a prescribed format from users whether it be searching for information (e.g., entering an airport on an airline ticketing service like www.travelocity.com.)

Problem Essence

• Memory- When entering details it’s possible the user may not remember the exact details that needs to be entered. Like in the case of an email address, the user will have a vague recollection of the exact email address, may be only the first few letters or the last name of the person they are sending the email to.

• Errors- In the case of email address or URLs it’s important that the entered details are correct. Typographical errors are common, and in some cases the input field will expect details in a prescribed format (e.g., fname.lname@domain.com) and in no other format.

How to fix possible errors even before the form is processed on the fly?
• *Time*- When such exact details are required to be filled, it will take users longer (number of steps or clicks) if they have to first look-up the input details outside of the workflow (e.g., filling out a form). For example, in an airline ticketing system, to determine the name of the airport, first the user will have to select the list of airports. Locate the airport code associated with a city, and then copy-n-paste it in. Furthermore, errors in the filling out the input field will hold up processing the form, and eventually make the process longer.

**Desired Goal (Functionality)**

Users need to enter details into a text form field *accurately* and *quickly* without having to rely solely on their *memory* for completing the task (i.e, make better use of knowledge-in-the-world).

**Solution Scenario**

The user wishes to email her friend John Doe, and opens the email compose window. As she begins to type “Do” in the TO field, she will see list populated with candidate matches. As the user types in more letters, the number of items in the list of suggestions/predictions that are extracted from the user’s address book will dwindle down. That is, if she types in DOE, and if the address book contains only one person named DOE, the list will disappear and the text field will be automatically populated, pending approval of the user. If there are more than one entries in the list, the user will hit the arrow key on the keyboard to accept the first entry in the list or tap on the down arrow key to select the next entry (i.e., if she was emailing PhillyCHI). Figure 5-18 shows two examples of such a feature.
Autofill shown as part of 1) an email client and 2) Firefox browser address bar

**Pros**

+ The *user will not have to remember* the exact phrase to be entered. In some cases, like for email addresses, the real email address may not be similar to the recipient’s name at all. With auto-fill, the user can start typing in the recipient’s name and the system will take care of connecting the name with a pre-defined format (i.e., email address).

+ Makes use of *information already available in the world* (e.g., address book) and can be easily adapted to be used in other domains like browser address bars (search browsing...
history and bookmarks or showing the airport codes (e.g., PHL for Philadelphia) on travel sites.

+ The suggestions *do not interfere* with the act of filling in the text field or disrupt the workflow by showing it below the input field, and

+ Helps *avoid typographical errors* in text input by predicting the intended input and offering them as suggestions and preventing delays in processing the form.

+ *Speed up* the inputting text by not having to completely type out the entire string.

**Cons**

- Requires *access to private user data* (usage behavior) in order to make meaningful suggestions (e.g., address book, browser history).

- For generic text fields (i.e., not custom fields like TO, BCC) effort must be made to *disambiguate the entries*. For example, typing in “CA” in a desktop wide search field may generate too many hits to be meaningful.

- Meaningful ordering or *relevance of suggestions* has to be determined based on the audience and task context (most visited vs. most frequently used) instead of adopting a one-size fits all approach.

- Pre-filling the text-field with the predicted first entry may be *distracting* to the user and get in the way of typing.

### 5.3 Discussion Summary

In this chapter we discussed the key findings related to structure of design advice and its impact on the perceived usefulness of design advice. We elaborated on cons (claim structure) affect the perceived usefulness of design advice, which also indirectly guided participants to consider more
unique number of design issues. In exploring the contribution of cons, we asserted that cons potentially reflect our prior experience applying said design advice, thus validating the applied nature of advice. We argued that design advice should be more than description of the solution along with pros and cons (i.e., more than claim-structure) especially in an artifact-driven approach to discovering best practices. Design advice should guide designers in managing trade-offs and understanding the implications (i.e., pros and cons). Trade-offs, we believe, reflect the true nature of design more closely, both from a process and resulting artifacts’ perspective. Abstracted trade-offs from existing solution(s) can serve as a new vocabulary for discussing design decisions, which echo the artifact-theory nexus proposal (Carroll & Kellogg). In such situations, articulating the problem becomes just as important for assessing the solution; problem framing mediates the process by which we arrive at solutions (Cross, 2004). Based on these findings we make a case for an enhanced structure, which combines elements of the pattern structure with the claim structure. Finally, we discussed how designers might use this structure as illustrated using the "accordion" and "autofill" advice examples. In the next chapter we outline briefly new areas of inquiry that will extend this study.
CHAPTER 6: CONCLUSIONS AND FUTURE WORK

This study compared the effectiveness of the pattern and claim structure on perceived usefulness, context of use described, rationale expressed and overall quality of interaction design advice. Based on a comparison of pattern structures in architecture and interaction design, an essence of a pattern structure was used as one of the candidate structures for articulating design advice. As part of the controlled study, effectiveness of the pattern-like structure was compared with claim-advice, which we found to closely resemble patterns in its intent and use. Participants were asked to articulate their design recommendations for the given design problem using a pattern or claim structure, which was rated on the four metrics outlined earlier. This rating data was used to answer the broad research question whether the structure used for preparing advice had an impact on the effectiveness or worthiness of advice.

As part of our study, we answered a call for much needed empirical work (Dearden and Finaly, 2000) surrounding patterns. Past empirical work, specifically controlled studies, assessed whether using a collection of design advice in the form of patterns had an impact on the quality of the designs created (Chung et al., 2004; Saponas et al., 2006; and Wania, 2008). The results from these did not find an impact of pattern-advice on the designed artifact. However, the interaction community has argued that a pattern format is a good way to capture and communicate design advice. The study presented here differs from past work by focusing exclusively on evaluating the impact of structure of design advice. Doing so allowed us to examine the contribution of the widely used pattern structure independent of the content of advice. Controlled studies in the past attempted to assess the content of advice, and implicitly, the structure of advice.

The structure used for communicating interaction design advice had an impact on the perceived usefulness of advice. And describing cons in design advice is potentially more useful
than just discussing pros or advantages of a design solution. We found that discussing pros and cons in design advice prompted designers to consider more unique design issues that are present in the context. Moreover, it offers a simple way to discuss and scope context of use. More importantly, we have argued that design advice can serve as a shared vocabulary for communicating what we know about design, which above all highlight trade-offs in context in the problem and solution. This new vocabulary resides at a level between concrete artifacts and theory and potentially represents how designers share and consume guidance in practice. Based on findings, we presented a new structure for articulating advice that placed trade-offs at the heart of the design problem (i.e., trade-offs that need to be made) and solution (i.e., trade-offs that have been made). This new structure suggests the inclusion of cons and description of the design problem in addition to the solution and discussion of fit (i.e., pros).

### 6.1 Limitations

As in any controlled study, this study also has limitations, which may be attributed to the following factors and need to be considered when interpreting findings: perceived usefulness of advice, experience preparing advice, and potential task description bias.

While we found that the structure affected the usefulness of advice, this should be regarded as perceived usefulness. It’s challenging to define an objective measure for usefulness because it may depend on the several factors. That is, it can depend on when advice is sought, how it is presented, and who consumes this. For example, design advice sought in the early stages of design may or should contain different kind of information than that sought at a later stage of implementation. Furthermore, the study used expert evaluators to judge the usefulness of advice, thus serving as surrogates for designers, and then basing usefulness on a specific design activity (room planner). While this may affect the generalizability of findings, it was a reasonable compromise in the interest of experimental rigor. Without having the same judge rating all the
advice produced by participants, it may have been difficult to compare across advice structures. And without at least two expert judges, it may be difficult to determine inter-rater (or judge) reliability. Using multiple expert judges here was akin to the peer-review process employed in academic venues.

The participant experience preparing advice and making a design selection from among alternatives may have also affected the outcome of the design activity. In the study, participants were required to make a design selection from a set of design alternatives. This was done in the interest of time and also to focus the participants on the core of the experiment, which was to prepare design advice. However, having to argue for a design alternative not sourced from personal inquiry or research may have been challenging for the participants. This may have to lead to lesser discussion in the design advice authored. And while the participants were selected from a population who had received formal instruction in HCI, not all of them may have been equally equipped to offer design advice, which is more than selecting a design alternative. However, the study implicitly benefited from this as it gave an opportunity to assess whether the structure guided this process of capturing advice.

And finally, both the expert judges and participants could have suffered from a task description bias. In other words, since both participants and judges received a brief task description, participants may have failed to account for explicitly in the advice prepared. Participants were instructed to prepare recommendation so as to be useful for readers who may not have access to the explicit task description. The expert judges may have also failed to note the presence or absence of task description in the design advice when rating advice since they too received the task description. This may be important because the inclusion of detailed task description may have influenced the context of use described rating and also may have given concreteness to the design recommendations. However, these limitations do not detract from
understanding the nature of design advice, and what it should represent. Neither does it away from the realization that describing cons can be useful aspect when communicating design advice.

6.2 Future Work

Knowing what we know now, the following sections briefly review new areas of research triggered by our findings. We also discuss the implications of our research as part of the three broad areas discussed below:

- Exploring patterns as a process to uncover “good” advice?
- The value of rationale in design advice as opposed to “good” designed instances
- Presenting design knowledge as a collection or as a language

6.2.1 Patterns as a Process to Uncover “Good” Advice

The word "design" can imply both the process by which an artifact is created and the artifact itself. Similarly, patterns too may be understood as a process (i.e., pattern process) and the result (design advice). When introducing the concept to patterns to the design community in his book, the TWoB, Alexander did not distinguish whether pattern is a process or result. Our interpretation is that the two concepts seem so closely related and sequenced, that they seem inseparable to some extent; patterns don't exist without a pattern process. In other words, patterns emerge from a discovery process, which is outlined as the pattern process (Alexander 2000, p10-15). Without going through this discovery process, it will be difficult to claim any advice is really a pattern-- a common/recurring "good" solution to a problem presented in context.

However, when defining what is a pattern, including Alexander's own definition, the process aspect of discovering a pattern may have got obscured or de-emphasized. Pattern, the resulting artifact, has by far received greater attention when we look at how a pattern is commonly defined (c.f., comparison of pattern definitions). And consequently, it seems that when it came to evaluating the empirical benefits of patterns, the resulting body of work too has
focused on whether pattern-based advice provide benefits during designing (Saponas, Wania et al.). What seems amiss, and not actively communicated, is the contribution of the discovery process for patterns.

Following up on Kellogg and Caroll's take on an artifact-theory nexus or artifact-driven design advice, we argued that design advice may potentially exist as a middle ground between theory and concrete design instances. Furthermore, when authoring design advice, we may leverage both existing theories and concrete examples. The study presented in this dissertation empirically evaluated the impact of structure used in communicating advice. And as part of the discussion of findings, an enhanced structure for capturing design advice was recommended. We should, however, acknowledge that having a meaningful structure alone does not imply "good" advice. Using an analogy, we can agree that a well-constructed box does not imply that the contents are good/desirable. The contribution of the box can only go as far as to safely transporting or protecting its contents, whatever it might be. In the same way, there is nothing to prevent "bad" design advice from being represented using the enhanced structure.

The pattern process may potentially be a way to discover "good" advice. "Good" being defined as advice that aids designers in resolving their design trade-offs while achieving the desired usability benefits. A pattern-based process, in essence, involves exploring existing interactions in interactive systems under similar contexts of use to identify desirable and undesirable interactions. And then identifying the components (trade-offs) that helps achieve this. But because good design by itself is invisible and relative, identifying "good" design IS the challenge. And consequently, capturing design advice that guides designers towards "good" design is also a challenge, but not impossible. The pattern process appears to be one way achieve this. For a more concrete instance of this process, Wania (2009) describes in greater detail how a pattern-based process was used to uncover interaction patterns in search interfaces, which then
lead to discovery of a pattern language for information retrieval systems. We see the pattern
process as a pre-cursor to capturing advice as claims, patterns or in the suggested enhanced
structure.

We believe exploring and evaluating the empirical benefits of "patterns as process" to be of
value to the interaction design community. We see the following questions to be of interest

- Does following a pattern-process result in design advice that engenders "good" design
  and consequently lead to better designed artifacts?
- If so, does a pattern-process replace or complement design methods in HCI?
- Under what conditions will a pattern-process are preferred? Does it depend on the
  application domain or the type of activity (design vs. evaluation)?

6.2.2 Capturing Guidance as Design Advice with Narratives or Design Instances
In the discussion section we outlined the desired components of design advice, which were the
following:

- Problem Setting
- Problem framing
- Design Problem
- Solution Scenario + design instances
- Pros
- Cons

We argued for why each of these sections add value to communicating design advice, and
provided two examples illustrating how this structure can be used to document advice. In the
discussion here, we are focusing on a distinction between design advice and design instances.
Design advice, as we have argued for it, contains design instances along with a narrative that
situates and explains the essential problem solved by the design instances. In other words, a
curated set of design instances with rationale describing why these are "good" and under which circumstances (design advice= design instance + narrative/rationale).

Constructing this narrative, using the enhanced structure, is a significant amount of effort on the part of someone capturing this advice, and that much more information a designer or reader may be forced to consume. While we can provide arguments in support of why this narrative is useful, as we have done so earlier, the empirical benefits of this extra content and effort is yet to be evaluated. If we are to consult pattern collections that have spawned on the web, we will find a mix of resources--one that captures design instances along with a narrative or explanation (www.welie.com), and those that simply include screenshots of the "good" design (e.g., www.patterntap.com) that are organized by user tasks or specific aspects of interaction. It seems that the screenshots approach seems to get a more active participation from the user community. But this may be because there are fewer obstacles to participate.
Figure 6–1 Comparing design guidance that include rationale with those which offer a set of “good” design instances; Breadcrumb Pattern (design instances) from http://patterntap.com/tap/collection/breadcrumbs; Breadcrumb Pattern (design instances + rationale) from http://www.welie.com/patterns/showPattern.php?patternID=crumbs

As shown in the comparison in figure 6-1., the name used to identify this advice, and potentially the design instances, are common to both representations. If we are to pursue exploring how to best capture design knowledge, it is important to decide what level of fidelity is enough for communicating and using this knowledge. In other words, is including rationale worth the effort. The questions stemming from this line of inquiry are the following:

- What is the value of capturing advice in the enhanced structure as opposed to curating good design instances?
- Which format is potentially more useful within the context of a design activity?
• Does the design lifecycle impact which format is more relevant? As in, are design instances more appropriate during early design discussions, and design advice (as narratives) appropriate for detailed design?

6.2.3 Design Advice as Collection or Language(s)?
When discussing the enhanced structure, we presented advice for AUTOFILL and ACCORDION. These are two specific interactions. But designers are often faced with problems of varying breadth and depth. That is, design problems may be about a specific interaction or be at a much larger scale or scope. For example, in design of a sales dashboard, choosing the correct graphical representation for highlighting sales targets in one kind of design problem; deciding the various components of a sales dashboard, what to show first, where to place which kind of information to be useful is a design challenge of a much larger scope. Correspondingly, we may need design guidance that match these varying scope of design problems. We see two areas where there are outstanding questions that need to be answered. First, given a specific interaction, for example AUTOFILL, how detailed does this advice needs to be without becoming prescriptive. Secondly, how to determine the appropriate scope (i.e., boundaries for guidance)?

A cursory inspection of available design advice, whether in the form of interaction patterns or claims, reveal a propensity towards collections instead of languages. The language concept in interaction design is an attempt to relate it to Alexander's work on a pattern language for design of living spaces (Alexander et al., 1977). A potential distinction between a language approach and a collection is that the relationship between pieces of design guidance is more explicit and present in languages. In a collection, the different pieces of design advice are loosely bound by the term offered to identify a collection. And this term may be indicative of a set of shared attributes amongst the different pieces of advice. Between the two approaches, a language approach appears to be more rigorous as it seems to expect a more precise relationship between
two pieces of advice. At least in the case of "A Pattern Language," the patterns use hierarchy of spaces as an organizing principle. That is, design guidance for a WINDOW SEAT follows and is related to design guidance for ALCOVES. When discussing interactive systems, an analog for hierarchy of spaces is not readily visible, but not completely absent either. For example, AUTOFILL can be argued as a good design practice when seeking user input in FORMS. We can look at a FORM as the broader context for AUTOFILL. In some sense this may represent a hierarchy of interaction or interactive spaces. Adopting this perspective, looking around, we can find other related pieces of advice that taken as whole may correspond with best practices for design of a FORM (e.g., IN-LINE VALIDATION, PROGRESS INDICATORS).

There are no clear guidelines that will enable designers capturing best practices to scope the guidance down to something as specific like AUTOFILL, and then determining how much detail should the guidance for AUTOFILL contain. For the later, while the current study did not focus on this explicitly, we found that that using the pros and cons approach appeared to offer a simple way to discuss relevant aspects of the advice and context. But this heuristic is yet to be evaluated. Furthermore, the current study focused on a single piece for design advice and did not address the questions related to scope directly. A related, and broader, empirical question is whether collections or languages are an effective way to capture and share design guidance. It's possible that a loose coupling between the difference pieces of design advice may be easier to manage, evolve and apply (i.e., as a collection). On the other hand, tightly coupled pieces of design advice presented as a language may be more comprehensible and potentially engender better quality designs.

We argue that patterns are a natural approach exploited by design professionals. However, the discussion on what patterns may have to offer has been subverted by the narrower focus on
patterns-as-result. As we have discussed above, patterns need a renewed focus and rethink, but this time exploring the benefits of patterns-process. We also asserted that trade-offs in context are integral to a design activity, and why design guidance needs to be about resolving trade-offs and not about specific usability goals. Furthermore, discussing cons as part of the advice can lead to identifying participating design issues, help in scoping the context-of-use, reflect the maturity and confidence in given advice, and potentially seed re-design activity. Design advice, as we have presented it (i.e., unifying patterns and claims), has the potential to fulfill a gap between theory based design methods and concrete design instances.
REFERENCES


Alexander, C. *Pattern language.com curriculum vitae*

Alexander, C. The origins of pattern theory, the future of the theory, and the generation of a living world. *Keynote Address to the Eleventh Annual Conference on Object-Oriented Programming Systems, Languages, and Applications,*


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


Collection of Pattern conferences, workshops and panels from to. *Collection of pattern conferences, workshops and panels from 1997 to 2004*

Computer Professionals for Social, Responsibility. *A pattern language for living communication*


Fincher, S. "Capture of practice": Is it obvious? Paper Presented at the BCS HCI Group/IFIP WG 13.2 Workshop on HCI Patterns, London,


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

Fincher, S., Finlay, J., Greene, S., Jones, L., Matchen, P., Thomas, J., et al. Perspectives on HCI patterns: Concepts and tools. CHI ’03: CHI ’03 Extended Abstracts on Human Factors in


Griffiths, R. N., Pemberton, L., & August. Don't write guidelines write patterns!


Hill, 347-356.

Hillside.net. Hillside.net


*Implementing a pattern library in the real world: A yahoo! case study - boxes and arrows: The design behind the design*


Leacock, M., Malone, E., & Wheeler, C. Implementing a pattern library in the real world: A yahoo! case study. *American Society for Information Science and Technology*


Martin, D., Rouncefield, M., & Sommerville, I. Applying patterns of cooperative interaction to work (re)design: E-government and planning. *CHI ’02: Proceedings of the SIGCHI*


McInerney, P. UI patterns: A workshop for designers. *CHI 2002 Workshop Report,*


Mullet, K. Structuring pattern languages to facilitate design. *CHI2002 Patterns in Practice: A Workshop for UI Designers,*


Nielsen, J. (2003). *Usability 101: Introduction to usability. alertbox*


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


Rittel, H., 1984, "Second-Generation Design Methods", in "Developments in Design Methodology", N. Cross (Editor), John Wiley & Sons, UK pp. 317-327


Good Practices to Patterns", (Helsinki, Finland, 2003).


Seffah, A., & Jawahery, H. On the usability of usability patterns ,ÄuWhat can make patterns usable and accessible for common developers,Äù. CHI 2002 Workshop on Patterns: Patterns in Practice,


The Brighton Usability Pattern,Collection. The brighton usability pattern collection

The Pedagogical Patterns, P. Pedagogical patterns

Tidwell, J. (2005). Designing interfaces: Patterns for effective interaction design O'Reilly Media,
Tidwell, J. *Common ground: A pattern language for human-computer interface design*

Tidwell, J. *UI patterns and techniques*


Tohidi, M., Buxton, W., Baecker, R., & Sellen, A. (2006). Getting the right design and the design right. *Proceedings of the SIGCHI conference on Human Factors in computing systems* (pp. 1243-1252)


van Welie, M., & Troettenberg, H. Interaction patterns in user interfaces. *Proceedings of the 7th Conference on Pattern Languages of Programs (PLoP)*.


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


Yahoo Design Pattern, L. *Yahoo design pattern library*

Yin, R. *Case study research: Design and methods* Sage.
APPENDIX A: PRE-TASK QUESTIONNAIRE

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. Gender............ Male / Female

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

4. What is your current major? ______________________________________

5. 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: ________________________________________________

6. 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

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

Thank you for completing this survey!
APPENDIX B: DESIGN TASK (PATTERN CONDITION)

TASK
A room planner allows shoppers to add products from the catalog to the room plan and try out different arrangements. It would help shoppers avoid the hassle of ordering furniture that is too big or too small for their homes.

You have to choose one among three menu layouts for presenting a product catalog on the room planner interface. You will be provided with the options. Explain the reasoning behind your choice using the provided instructions. Address the following questions when explaining your decision:

a. What is the user trying to do?
b. How will the recommended menu option be used?
c. Explain why it is the best choice in this given situation?

Explain your decision as if you are teaching someone else to make a similar choice. Others reviewing your recommendation may not see the options that you rejected. Mark (a), (b) and (c) in the body of your explanation.

Figure 1. An example room planner interface showing the main task area. Each object shown in this task area can be moved or rotated.
APPENDIX C: DESIGN TASK (CLAIM CONDITION)

TASK
A room planner allows shoppers to add products from the catalog to the room plan and try out different arrangements. It would help shoppers avoid the hassle of ordering furniture that is too big or too small for their homes.

You have to choose one among three menu layouts for presenting a product catalog on the room planner interface. You will be provided with the options. Explain the reasoning behind your choice using the provided instructions. **Address the following questions when explaining your decision:**

a. How will the recommended menu be used?
b. Advantages of using your design choice
c. Disadvantages of using your design choice

Explain your decision as if you are teaching someone else to make a similar choice. Others reviewing your recommendation may not see the options that you rejected. **Mark (a), (b) and (c) in the body of your explanation.**

**Figure 1.** An example room planner interface showing the main task area. Each object shown in this task area can be moved or rotated.
APPENDIX D: DESIGN TASK (CONTROL CONDITION)

**TASK**

A room planner allows shoppers to add products from the catalog to the room plan and try out different arrangements. It would help shoppers avoid the hassle of ordering furniture that is too big or too small for their homes.

You have to choose one among three menu layouts for presenting a product catalog on the room planner interface. You will be provided with the options. Explain the reasoning behind your choice using the provided instructions.

Explain your decision as if you are teaching someone else to make a similar choice. Others reviewing your recommendation may not see the options that you rejected.

*Figure 1.* An example room planner interface showing the main task area. Each object shown in this task area can be moved or rotated.
**APPENDIX E: MENU OPTIONS**

<table>
<thead>
<tr>
<th>(A) Directory-Tree</th>
<th>(B) Drop-Down</th>
<th>(C) Accordian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expand and collapse different sections using +/−</td>
<td>Drop down reveals the product categories (down arrow)</td>
<td>Clicking on the arrows expands/closes the section.</td>
</tr>
<tr>
<td>Multiple sections can be opened at a time</td>
<td>Only one section open at a time</td>
<td>Only one section open at one time.</td>
</tr>
<tr>
<td>Product information and image displayed inside the menu</td>
<td>Product information and image displayed inside the menu</td>
<td>Product information and image displayed inside the menu</td>
</tr>
<tr>
<td>Fixed height for the menu</td>
<td>Fixed height for the menu</td>
<td>Fixed height for the menu</td>
</tr>
<tr>
<td>Scroll (outside) to view the entire open catalog</td>
<td>Scroll (inside) to view more products</td>
<td>Scroll (inside) to view more products</td>
</tr>
</tbody>
</table>
APPENDIX F: PARTICIPANT WORKSHEET (PATTERN CONDITION)

Explain the reasoning behind your choice using the provided instructions. Address the following questions when explaining your decision:

a. What is the user trying to do?

b. How will the recommended menu option be used?

c. Explain why it is the best choice in this given situation?

Explain your decision as if you are teaching someone else to make a similar choice. Others reviewing your recommendation may not see the options that you rejected. Mark (a), (b) and (c) in the body of your explanation.

<table>
<thead>
<tr>
<th>RANK</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX G: PARTICIPANT WORKSHEET (CLAIM CONDITION)

Explain the reasoning behind your choice using the provided instructions.

Address the following questions when explaining your decision:
   a. How will the recommended menu be used?
   b. Advantages of using your design choice
   c. Disadvantages of using your design choice

Explain your decision as if you are teaching someone else to make a similar choice. Others reviewing your recommendation may not see the options that you rejected. Mark (a), (b) and (c) in the body of your explanation.

<table>
<thead>
<tr>
<th>RANK</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX H: PARTICIPANT WORKSHEET (CONTROL CONDITION)

Explain the reasoning behind your choice as if you are teaching someone else to make a similar choice. Others reviewing your recommendation may not see the options that you rejected.

<table>
<thead>
<tr>
<th>RANK</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>#2</td>
<td>#3</td>
</tr>
</tbody>
</table>


APPENDIX I: POST-TASK QUESTIONNAIRE (PATTERN & CLAIMS CONDITION)

Thank you for your participation.

Please place an X in the box that expresses your level of agreement or disagreement with the following statements.

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neither agree nor disagree</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>I understood the structure provided in steps a, b, and c to explain my design recommendation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I believe the structure helped me explain my design recommendation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I would use the structure in step 1 in future for explaining interface design recommendation or advice</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I understood what I was asked to do.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I enjoyed this activity.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Please answer the following questions.

What did you like about this activity?

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

What did you dislike about this activity?

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
APPENDIX J: POST-TASK QUESTIONNAIRE (CONTROL CONDITION)

Thank you for your participation.

Please place an X in the box that expresses your level of agreement or disagreement with the following statements.

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neither agree nor disagree</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>I understood what I was asked to do.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I enjoyed this activity.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Please answer the following questions.

What did you like about this activity?

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

What did you dislike about this activity?

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________
APPENDIX K: FLYER INTRODUCING STUDY

DREXEL UNIVERSITY
COLLEGE OF INFORMATION SCIENCE & TECHNOLOGY

EVALUATING THE IMPACT OF A PATTERN STRUCTURE ON COMMUNICATING INTERACTION DESIGN ADVICE

This is a great opportunity to help us learn more about how to communicate interface design knowledge and design recommendations! We are seeking volunteers to participate in a study that will help us understand how to capture and deliver interaction design advice.

Participants will receive $XX for their participation

Participants will be asked to choose one interface design option from a set of given options. They will document the reasons behind their choice using a given format. Participants will also be expected to answer questions related to their experience using and designing systems.

It will take approximately 1 hour to complete this exercise. If you are interested in participating in this study please contact:

George Abraham
(215) 895-5912
George.Abraham@ischool.drexel.edu
Rush Building #106 (iCommons)
APPENDIX L: REFERENCE SHEET FOR EVALUATING RECOMMENDATIONS

Quick Summary of Metrics

1. CONTEXT OF USE DESCRIBED
   What is the task and task context like?
2. USEFULNESS OF DESIGN ADVICE
   Why does menu choice work for the task context (coverage of relevant issues)?
3. RATIONALE OFFERED
   What are the reasons for calling this a good fit? Any implications?
4. OVERALL QUALITY OF ADVICE
   Compared to other recommendations in the sample, how good is this one?

Metrics Explained With Examples

1. CONTEXT OF USE DESCRIBED

Looking for concreteness and detail described when discussing the menu option (i.e. task + keywords/phrases + usage scenario)

<table>
<thead>
<tr>
<th>NO DESCRIPTION</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>AS GOOD AS IT GETS</th>
</tr>
</thead>
</table>

A good description will avoid using jargon (e.g., efficiency, clarity, simplicity). It will touch upon aspects beyond those explicitly stated in the menu description (c.f. task handout menu comparison table). In the good example, shown below, the description refers to pairing up furniture items like rugs and sofas instead of referring to these as product categories. Even when discussing menu aspects like a +/- buttons on Directory Tree, good descriptions will refer to it as part of a usage scenario.

- ✔️ Often when shopping for furniture, design-savvy shoppers may want to coordinate their furniture picks (i.e., pieces that go together). For this, it may be useful to allow people to peruse through different product categories at the same time. Clicking the +/- sign in the directory-tree menu will allow users to have control over which category they will like to see in detail, without having to open and close categories all the time. For example, when picking “rugs,” they may like to keep “sofas” also open to color-coordinate.

- ✗ I will recommend the drop-down menu because it’s intuitive to users. Also the menu structure shows categories that keep the product catalog organized and clutter free. Having an organized product catalog can prevent confusion for the users. The drop-down is a simpler design choice unlike the other dynamic options.

2. USEFULNESS OF DESIGN ADVICE

Measures how much the expert judges agree with the relevance of issues mentioned given the context of use.
A good recommendation draws attention to problems or issues that arise between the interface elements and the context of use described. In other words, talking about a fit between context of use and the design. Also, there may be little value in describing benefits of a design choice that is based on an assumption (i.e., a menu being more intuitive). Please note that the three different menu options used in this experiment differed slightly from each other. What they shared was the context-of-use. In reality, the product categories will be numerous (>5), and there are also some physical restrictions imposed on the interface (e.g., menu height is fixed).

- ✔ The Accordion menu is a better choice here because the menu height is fixed here. A directory-tree menu can become unwieldy very quickly. The limited menu height will push the category names off the viewable area or screen fold, hiding the product categories. Given the large number of products usually available in furniture catalogs, opening multiple categories open, like in a tree menu, can force a lot of scrolling without a clear indication when a category begins or ends.

- ✗ I will recommend the Directory-tree menu because the +/- is more intuitive to users. Since multiple categories can be opened, the shoppers can pick the products they like more quickly. They can expand all the options and peruse the entire catalog by using a single scroll feature on the outside of the menu, which is easier to use than scrolls placed inside the menu like in the other menus.

3. RATIONALE OFFERED;

How well does the advice communicate reasons behind the design choice and it’s implications (i.e., persuasiveness)

A good recommendation will consider more number of issues related to the design choice, the user, the context of use, and their interaction. At least five design features were described for each design choice, placed side-by-side for easy comparison. Ideally, the participants should have touched upon the implication of each design feature within the given the task context. In the data collected, it will be common to see participants reference specific aspects of interaction (e.g., flexibility, memory, user-experience, design conventions, time, number of clicks). Good advice addresses more than three issues. Poor advice only addresses one or two issues without providing a persuasive explanation.

- ✔ The Accordion menu is a better option given the fixed menu height constraint. It serves two functions simultaneously: keep product categories visible at all times, allows shoppers to drill down within each category. By keeping product categories visible, the users do not have to remember the categories nor do they need to actively check which section they are in. The room planner may used by users with varying degrees of experience both using computer interfaces in general or people who have used other room planners. The Accordion limits the number of categories to one, and help in maintaining the shoppers focus on task at hand without overwhelming them with long lists, suited to both experienced and inexperienced users.
• I recommend the *directory-tree* menu because it will allow shoppers to go directly to the category they like and allow them to select products from multiple categories by control clicking. They can also compare products across categories.

4. **OVERALL QUALITY OF ADVICE**: overall, how good is the design advice provided?

   ![Rating Scale](Very Poor: 1 2 3 4 5 6 7 Excellent)

   While the metrics defined in criteria 1, 2, and 3 may suffice; this option attempts to elicit a subjective response from the expert judges based on their experience evaluating design recommendations.

   **NOTE:**

   How to interpret and rate when participants adopt a comparison of menus strategy in their recommendation instead of discussing their reasons for picking a particular menu option?

   Although participants were given explicit instructions to avoid discussing menu options they did not pick, some have used a compare-and-contrast strategy to explain their choice. For example, a participant writes, “The drop-down menu hides the product categories unlike the accordion menu.” In such cases, it will be fair to interpret this comment as highlighting a benefit of the accordion menu (i.e., product categories are always visible when shopping in the accordion style menu).
APPENDIX M: EVALUATION SHEET WITH RECOMMENDATION

ACCORDION

The user is trying to find items to fit within their room(s) and probably have an idea of what type of items they are looking for already. They are trying to view available items, not necessarily just browsing to see what IKEA has to offer in general.

Considering this, the accordion-style menu provides the quickest way to jump between sections while still maintaining a clear list of options. The user does not see any items not in that section (as they would in directory-tree) so they will not be influenced by alternatives. The drop-down style requires users to look through the sections before being able to view any items. A scroll bar within the menu allows users to look through the possible options and may be better than listing all options for each section. Additionally, users may find the Directory-Tree style menu becomes too cluttered when all the sections are open and can make browsing through tedious (depending on # of items listed).

Users are coming to this site in order to save time when shopping for furniture. By providing them with clear sections and only one at a time, the accordion menu/Third menu allows users to look for each item they want. The drop-down style could cause a longer search time for users if they are not sure where an item might be listed and having to go through each section. Additionally, the directory-tree style may confuse less savvy users because they cannot easily locate the sections they previously had open.

<table>
<thead>
<tr>
<th>Context of Use Described</th>
<th>No Description</th>
<th>As Good As It Gets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 2 3 4 5 6 7</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Usefulness of Design Advice</th>
<th>Not Relevant</th>
<th>Extremely Relevant</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 2 3 4 5 6 7</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rationale Offered</th>
<th>None</th>
<th>As Good As It Gets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 2 3 4 5 6 7</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Overall Quality</th>
<th>Extremely Poor</th>
<th>Excellent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 2 3 4 5 6 7</td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX N: “GOOD” EXAMPLE

(i.e., usable, useful, includes rationale, and overall good advice)

The Accordion style menu is a good choice given the restricted the height of the menu and multiple products and categories. This menu will be used to present the product catalog on the room planning interface. Users can drag and drop the products from the catalog to the room plan. Given the limited menu height, and the traditionally high number of product available in catalogs, opening multiple categories in the options like directory menu will push the category names off the viewable area or screen fold, hiding the product categories. It’s also relatively simple to interact with this menu because clicking on the up and down arrow will open a category. On click a previously open category will automatically collapse keeping the list manageable. For example, if you click on “lighting” when viewing “storage,” “storage” will collapse and “lighting” will expand. All the product categories will be visible at all times so users don’t have to remember where they are in the catalog. This interaction is simple enough for web-savvy users, and also for average users without overwhelming them with choices. Given the task of room planning, it is only natural to mix and match furniture pieces. Here, the user can select a product category they like, pick the product type (e.g., sofa) and drag and place it on the floor plan. Since the image shown on the room plan is representative, one may not need to remember what color or type of furniture that was picked.

[1] USABILITY OF DESIGN ADVICE (Discusses a scenario and avoids jargon)

NO DETAIL < 1 2 3 4 5 6 7 > AS GOOD AS IT GETS

[2] USEFULNESS OF DESIGN ADVICE (Addresses constraints imposed by the task context)

NOT RELEVANT AT ALL < 1 2 3 4 5 6 7 > EXTREMELY RELEVANT

[3] RATIONALE OFFERED (Argues for the choice by explaining more than three issues)

NONE < 1 2 3 4 5 6 7 > AS GOOD AS IT GETS

OVERALL QUALITY OF ADVICE (Discusses a scenario and avoids jargon)

VERY POOR < 1 2 3 4 5 6 7 > EXCELLENT
APPENDIX O: COMPARING ADVICE STRUCTURE
<table>
<thead>
<tr>
<th>Pattern Structure</th>
<th>Claim Structure</th>
<th>Enhanced Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>N/A</td>
<td>PROBLEM SETTING: The interface consists of an empty staging area or sand-box like a room planner, where objects can be placed and manipulated. A collection of objects consisting of images and descriptions is available to choose from (e.g., symbols, shapes, or furniture items) for adding to the stage area. Objects (e.g., flowchart symbols) can be added or removed on this interface.</td>
</tr>
</tbody>
</table>
| N/A               | N/A            | PROBLEM ESSENCE:  
  • *Visibility* - Users needs to be able to see an overview of objects/shapes available to them. If absent, it user may not be aware that these are available, and also forces the users to remember the options available to them. It is preferred that this collection of objects is conveniently located in one place instead of scattering them all over the interface.  
  • *Complexity* - when faced with a large number of objects, users may get overwhelmed with the variety or number. The challenge is to present these |
options in an organized manner such that users can easily switch between an overview and detail views.

- **Spatial Constraints**: On the interface, the staging area gets visual/spatial priority, and consequently, the amount of space available to show the collection of objects is limited. The collection of objects needs to be displayed without obscuring the main activity, for example, staging a room.

<table>
<thead>
<tr>
<th><strong>DESIGN PROBLEM:</strong> Users need to be able to <strong>view</strong> and <strong>select</strong> from a large collection of objects shown to them in a <em>familiar and unobtrusive</em> manner. They should be able to switch between overview and detail views <strong>quickly</strong> to avoid being overwhelmed while making the best use of available visual <em>space</em> to display the collection.</th>
<th>N/A</th>
<th><strong>DESIGN PROBLEM:</strong> Users need to be able to <strong>view</strong> a large collection of objects shown to them in a <em>familiar and unobtrusive</em> manner. They should be able to switch between overview and detail views <strong>quickly</strong> to avoid being overwhelmed while making the best use of available visual <em>space</em> to display the collection.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SOLUTION:</strong> The user wants to construct a flowchart of a business process. The main drawing or charting area is empty. On the left side of the staging area, the user sees a vertical list of item groupings with each group visually separated from each other by a heading name. For an accordion menu (refer figure XX-1), one of these categories will remain in an expanded state revealing</td>
<td><strong>SOLUTION:</strong> The user wants to construct a flowchart of a business process. The main drawing or charting area is empty. On the left side of the staging area, the user sees a vertical list of item groupings with each group visually separated from each other by a heading name. For an accordion menu (refer figure XX-1), one of these categories will remain in an expanded state revealing</td>
<td><strong>SOLUTION:</strong> The user wants to construct a flowchart of a business process. The main drawing or charting area is empty. On the left side of the staging area, the user sees a vertical list of item groupings with each group visually separated from each other by a heading name. For an accordion menu (refer figure XX-1), one of these categories will remain in an expanded state revealing</td>
</tr>
</tbody>
</table>
the object thumbnails within it. Each group heading is visually differentiated from the items. A visual thumbnail of the items is shown instead of using text to describe the item. Once the users locate an object/shape required, they select and drag it onto the drawing area. Selecting or clicking on any other group headings (i.e., ones not expanded) will collapse the previously open category and open the new selection. At any one point, only contents from a single group are shown.

<table>
<thead>
<tr>
<th>RATIONALE:</th>
<th>PROS:</th>
<th>PROS:</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ The group headings give a visual cue to the users about the type of objects available to them. The visual thumbnails of objects promote recognition instead of reading item names available. + Since only one category is available at a time, the user will not be overwhelmed with the number of choices, thus potentially reducing complexity of the interface. They can focus on one category at a time. + If an activity-centric grouping strategy is used, the user will not need to waste time switching between multiple categories. What they need to complete the activity is shown in one place while hiding items not relevant. Neither will...</td>
<td>+ The group headings give a visual cue to the users about the type of objects available to them. The visual thumbnails of objects promote recognition instead of reading item names available. + Since only one category is available at a time, the user will not be overwhelmed with the number of choices, thus potentially reducing complexity of the interface. They can focus on one category at a time. + If an activity-centric grouping strategy is used, the user will not need to waste time switching between multiple categories. What they need to complete the activity is shown in one place while hiding items not relevant. Neither will...</td>
<td>+ The group headings give a visual cue to the users about the type of objects available to them. The visual thumbnails of objects promote recognition instead of reading item names available. + Since only one category is available at a time, the user will not be overwhelmed with the number of choices, thus potentially reducing complexity of the interface. They can focus on one category at a time. + If an activity-centric grouping strategy is used, the user will not need to waste time switching between multiple categories. What they need to complete the activity is shown in one place while hiding items not relevant. Neither will...</td>
</tr>
<tr>
<td>Pros</td>
<td>Cons</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>They need to spend time to collapse a previously open category as clicking a new category automatically collapses the previous one.</td>
<td>The organization of objects or shapes needs to be evaluated to determine whether to group them by type or activity, which in turn depends on the task context. If it’s a diagramming application then the logical grouping may be based on type of object (e.g., callouts, basic shapes). Another possible grouping may be according to the activity (e.g. flow-chart, living room) such that shapes required to complete a certain activity are grouped together. Depending on the scope of the activity, the objects needed may belong to different categories (e.g., matching colors or style of furniture). This may require the user to keep more than one category open, which is not possible with the accordion menu structure.</td>
<td></td>
</tr>
<tr>
<td>In situations when the total menu height is fixed, the accordion structure balances visibility of group headings with details of the categories. The accordion makes best use of the available area minus the space used to display the group headings.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>In situations when the total menu height is fixed, the accordion structure balances visibility of group headings with details of the categories. The accordion makes best use of the available area minus the space used to display the group headings.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

N/A

CONS:
- The organization of objects or shapes needs to be evaluated to determine whether to group them by type or activity, which in turn depends on the task context. If it’s a diagramming application then the logical grouping may be based on type of object (e.g., callouts, basic shapes). Another possible grouping may be according to the activity (e.g. flow-chart, living room) such that shapes required to complete a certain activity are grouped together. Depending on the scope of the activity, the objects needed may belong to different categories (e.g., matching colors or style of furniture). This may require the user to keep more than one category open, which is not possible with the accordion menu structure.
A large numbers of groups or headings (and menu height is fixed) may use up a significant amount of the vertical space. The remaining space may be too little to display the item details or to be useful to the users.