User-Centered Critical Parameters for Design Specification, Evaluation, and Reuse:
Modeling Goals and Effects of Notification Systems

Christa M. Chewar

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Applications

July 15, 2005
Blacksburg, VA

D. Scott McCrickard, Chair
Doug A. Bowman
John M. Carroll
Deborah S. Hix
Alistair G. Sutcliffe

Keywords: Human-computer interaction, Usability engineering, Information design

User-Centered Critical Parameters
for Design Specification, Evaluation, and Reuse:
Modeling Goals and Effects of Notification Systems

Christa M. Chewar

(ABSTRACT)

Responding to the need within the human-computer interaction field to address ubiquitous and multitasking systems more scientifically, this research extends the usefulness of a new research framework for a particular class of systems. *Notification systems* are interfaces used in a divided-attention, multitasking situation, attempting to deliver current, valued information through a variety of platforms and modes in an efficient and effective manner. Through review of literature and experiences with empirical dual-task perceptual studies, we recognize a lack of unifying framework for understanding, classifying, analyzing, developing, evaluating, and discussing notification systems—fundamentally inhibiting scientific growth and knowledge reuse that should help designers advance the state-of-the-art.

To this end, we developed a framework (referred to as the *IRC framework*) for notification systems research based on a core taxonomy of critical parameters describing user goals. Next, we extend the framework, focusing on three key aspects: **1) a system description process,** allowing articulation of abstract design objectives that focus on critical user requirements; **2) interface usability evaluation tools,** enabling comparison of the design and user's models, while supporting generalizability of research and early identification of usability concerns; and **3) design comparison and reuse mechanisms,** saving time and effort in requirements analysis and early design stages by enabling design reuse and appreciation of design progress.

Results from this research include the development of tools to express IRC design models (*IRCspec*) and user's models (*IRCresults*), and the extension of the critical parameters concept. Validation studies with novice designers show sufficient assessment accuracy and consistency. Leveraging these tools that help designers express abstract, yet critical, design intentions and effects as classification and retrieval indices, we develop a repository for reusable design knowledge (*a claims library*). Responding to challenges of design knowledge access that we observed through initial user testing, we introduce a vision for an integrated design environment (*LINK-UP*) to operationalize the IRC framework and notification systems claims library in a computer-aided design support system. Proof-of-concept testing results encourage the thought that when valuable design tools embody critical parameters and are coupled with readily accessible reusable design knowledge, interface development will improve as a scientific endeavor.

ACKNOWLEDGEMENTS

I would first like to extend my deep appreciation to my advisor, Dr. Scott McCrickard. From my very first days as a graduate student at Virginia Tech, Dr. McCrickard inspired me to produce great research and helped me develop the skills and confidence necessary to become his colleague. My initial research goals did not extend beyond a masters thesis, but he first encouraged me to aim higher and pursue a Ph.D. He taught me how to write academic papers, run experiments, and supervise research. He helped me understand the research landscape of our discipline, providing my foundation for lifelong learning. The climate Dr. McCrickard has established within his lab, that of academic excellence, integrity, teamwork, and fellowship, provided the perfect environment in which to conduct my research. To repay his service, I can only strive to provide mentorship of such caliber to my own future students.

I also wish to thank my committee members for their efforts. Each of them provided suggestions and encouragement that ultimately made my work better and clarified directions for future research. In particular, I am very grateful for Dr. Sutcliffe's extensive review of my written work. All of my committee members have been inspirational to me with their own professional contributions.

As an active-duty officer in the U.S. Army, my research time and this work was made possible through a fellowship under the U.S. Armys Advanced Civilian Schooling program. Helping me secure approval to pursue a Ph.D was Colonel Eugene Ressler, Major Gregory Conti, and Colonel Andre Sayles, all from the Department of Electrical Engineering and Computer Science at the United States Military Academy. Without their support and encouragement, I would not have had the resources necessary to complete this research.

Certainly, I have my family and friends to thank for all the support and encouragement they provided to me along the way. In particular, my parents nurtured my early love for reading and always provided the best educational opportunities. The confidence they have always had in me, particularly in my decision to pursue a Ph.D, has been a continuous source of motivation. Lisa and Scott McCrickard provided many of the extracurricular essentials in top-fashion: a "home away from home, emotional support, and a source of interesting leisure activities.

Throughout all phases of my work, the other students in our lab made great contributions—I have noted specific assistance throughout this document, in context. However, I am especially appreciatory of the frequent brainstorming sessions that also involved Ali Ndiwalana and Jacob Somervell (who were both always there with a helping hand), as well as the technical assistance provided by Edwin Bachetti. Having such a close-knit group made the research process fun. I sincerely hope this research climate will continue, and I wish the best of luck to the current and future students continuing research within this general line of work.

Contents

ABSTRACT ii

ACKNOWLEDGEMENTS iii

1 INTRODUCTION 1
1.1 Motivation & Vision 3
1.2 Goals and Objectives 3
1.3 Anticipated Impact 4
1.4 Overview of the Dissertation 5

2 RELATED WORK 7
2.1 Human-Computer Interaction and Interface Design 7
2.2 Notification Systems in Literature and Use 8
2.3 Evaluating Usability of Notification Systems 10
2.3.1 Preserving attention demand 11
2.3.2 Maximizing attention utility 12
2.4 Improving Usability of Notification Systems 13
2.4.1 Usability engineering approaches 13
2.4.2 Focusing research efforts 15
2.4.3 Software and knowledge reuse 18

3 BACKGROUND WORK: EVALUATING SECONDARY DISPLAY ATTRIBUTES 20
3.1 Effectiveness of Visual Attributes for Secondary Displays 20
3.1.1 Experiment 1: Position, color & area 21
3.1.2 Experiment 2 & 3: Other hues and visual attributes 29
3.2 Design Knowledge for Design Process? 34
3.2.1 Archiving study results as a claim 34
3.2.2 Limitations of empirical design research 36

4 PRELIMINARY WORK: THE IRC FRAMEWORK 37
4.1 Toward Unifying Descriptions of Notification Systems 38

4.1.1 A taxonomy of usability concerns 38
4.1.2 Attention-utility theme 41
4.2 A Design Space for Notification Systems 43
4.2.1 Critical model parameters 43
4.2.2 The IRC classification framework 47
4.2.3 Notification action models 49
4.3 Observed Benefits 53
4.3.1 Classifying existing systems 53
4.3.2 Interface testing and reengineering 55
4.3.3 Teaching undergraduates about notification design 61
4.4 Anticipated Utility 61
4.5 The Way Forward: A Complete Research Contribution 62

5 PROVIDING DESIGN MODEL IRCs TO EXPRESS REQUIREMENTS 64
5.1 Issues with Expressing IRC Design Models 65
5.1.1 Problem statement and general approach 66
5.1.2 Expected results and successful endstate 67
5.2 Do Notification Designers *Really* Manage IRCs? 67
5.2.1 Method: Analysis of design rationale 67
5.2.2 Professional and novice designer results 69
5.3 IRCspec Tool Design and Tuning Process 72
5.3.1 Tool design considerations 72
5.3.2 Tuning and retuning the underlying parameter values 77
5.3.3 Putting it all together: The IRCspec tool and system integration 78
5.4 Validation of IRCspec Consistency and Accuracy 79
5.4.1 Hypothesis, method, and procedures 79
5.4.2 Analysis and results 81
5.4.3 Study implications 84
5.5 Conclusions and Future Work 85

6 OBTAINING USER'S MODEL IRCs FROM USABILITY TEST RESULTS 87
6.1 IRCresults Motivation 87
6.1.1 Problem statement and general approach 90
6.1.2 Expected results and successful endstate 90
6.2 Issues with Using Critical Parameters 91
6.2.1 Critical parameters can... create scenario families 92
6.2.2 Critical parameters can... form a general design space 92
6.2.3 Critical parameters can... express problems 93
6.2.4 Critical parameters can... support mediated evaluation 93
6.2.5 Residual issues 94

6.3 A Generic Evaluation Tool for User's Model IRC Ratings 95
6.3.1 Relating abstract and concrete terms with equations 95
6.3.2 Intended use and evolution 101
6.3.3 Two methods for obtaining IRC variables in usability evaluations 103
6.4 Study 1: Analytically Evaluating Redesigns of The Scope 104
6.4.1 Testing and analysis procedure 105
6.4.2 Study results and implications 109
6.4.3 Broader implications: Comparison and reuse 110
6.5 Study 2: Empirically Evaluating New Expedia Notification Interfaces 112
6.5.1 Notification data and interfaces 112
6.5.2 Testing and analysis procedure 113
6.5.3 Broader implications: A full spectrum of usability data 118
6.6 Conclusions 119

7 REUSABLE DESIGN KNOWLEDGE WITH CRITICAL PARAMETERS 121
7.1 Initializing a Claims Library 122
7.1.1 Motivation 122
7.1.2 General approach and expected results 123
7.1.3 Developing the Notification Claims Library 124
7.1.4 Formative testing 132
7.1.5 Discussion and future work 137
7.2 Using the IRC Framework in an Integrated Design Process 139
7.2.1 Assessing the status quo 139
7.2.2 A broader vision: LINK-UP 141
7.2.3 Module #1, Requirements Analysis 144
7.2.4 Module #3, Participatory Design Negotiation 146
7.2.5 Module #4, Analytical Evaluation 148
7.2.6 Module #5, Empirical Lab-Based Testing 151
7.2.7 A Sixth Module? – Exploring Claim Relationships 153
7.3 General Discussion 156

8 INTEGRATIVE CASES, CONCLUSIONS, AND FUTURE WORK 158
8.1 Integrative Case Studies 159
8.1.1 Case Study 1: Designing an IVIS input method 160
8.1.2 Case Study 2: Designing a news notification system 163
8.1.3 Case Study 3: Designing a notification system for ticket prices 165
8.2 General Conclusions and Contributions 168
8.3 Directions for Future Work 170
8.3.1 Educating novice designers with LINK-UP 170
8.3.2 Synthesizing multidisciplinary research results 171

8.3.3 Advancing notification systems design with a reference task agenda 172
8.3.4 Narrowing the HCI–Software Engineering gap 172
8.4 Closing 173

REFERENCES **174**

APPENDICES **185**

A IRCspec Documentation, Testing Materials, and Results **186**
A.1 Design Rationale Analysis 186
A.2 Algorithm, Screenshots, and Program Code 192
A.3 Walkthrough Results 208
A.4 Experiment Materials 209
A.4.1 Design Problems 209
A.4.2 Manual IRC Calculation Instructions 213
A.4.3 Response Forms 214
A.5 Experiment Data and Analysis 216

B IRCresults Documentation, Testing Materials, and Results **223**
B.1 Equation Variables and Behavior 223
B.2 Experiment Materials 231
B.3 Analytical Experiment Data and Analysis 237
B.4 Empirical Experiment Data and Analysis 243

C Claims Library and LINK-UP Materials **251**
C.1 Claims Library Screenshots and Documentation 251
C.2 LINK-UP Screenshots and Pilot Test Results 259
C.2.1 Requirement Analysis Module screenshots 259
C.2.2 Analytic Module screenshots 264
C.2.3 Empirical Testing Module screenshots 272
C.2.4 Selected pilot testing results 274

List of Figures

3.1 *Attribute scales and encoding schemes used in the experiment are shown. Relative increases within attribute values are uniform. Subjects are shown the applicable scale before the start of each round.* 24

3.2 *Each of the three attribute images (Color, Area, Position) encodes the same dataset values 1, 5, 2. In a given round, one of these images is presented as a secondary task while the participant plays a game. Only three values are shown here, but the experiment's images encode ten.* 24

3.3 *At the start of each round, participants are shown the encoding scale (a), and then asked a question (b). Questions require min/max identification, ratio determination, or counting comparison results. Depending on the round type, viewing images (c) may or may not occur while playing a game (d). The game requires a high degree of attention to catch quickly falling blocks. At the end of the round, the question answering dialog (e) captures relative effectiveness of the image encoding and attention-division conditions.* 25

3.4 *Answer correctness by attribute type for levels of acceptable degradation. Note that levels of acceptable degradation cumulate from the left side of the figure (i.e., subjects that meet five percent acceptable degradation include those at zero through four percent, as well as five percent.* 26

3.5 *Percentage of correct answers for each attribute type, at increasing levels of primary task acceptable degradation.* 27

3.6 *Solid trendlines show that levels of significance for attribute ordering vary with degree of acceptable primary task degradation. In rounds with minimal primary task degradation (no more than 23%), subjects answered more questions correctly with position than color images, although both allowed more correct answers than area images.* 28

3.7 *Each chart shows information about a single question type. Left y-axis and histogram data show sample size distribution at various levels of primary task performance changes. Right y-axis and trendlines (2-period moving averages) show variation of answer correctness according to primary task performance changes.* 28

3.8 *The eleven attribute scales used in the three experiments, depicted in scale to each other. Relative changes between values (positions, angles, lengths, luminescence values, etc) are kept uniform.* 31

3.9 *Ordering of attributes for supporting communication of secondary information (indicated by answer correctness). Does not consider effect on primary task performance.* 32

3.10 *Ordering of attributes for sustaining primary task performance (indicated by change in game performance while image was visible). Does not consider support for communication of secondary information.* . . . 33

4.1 *The IRC framework, depicting notification systems categorizations according to blend of design model objectives (representing user goals) of interruption (I), reaction (R), and comprehension (C) axes, simplified: low (0) or high (1) and forming a notional cube.* . . . 47
4.2 *A human information processing stage model, from [135].* . . . 50
4.3 *Design model flows through the human information processing stage model (see Figure 4.2) for each of the eight main notification systems IRC categorizations (see Figure 4.1), providing a diagnostic tool for thought. Noting unique design path trajectories for each categorization can help identify potential bottlenecks in the attentional flow and progression through the Stages of Action [100].* . . . 52
4.4 *Plots of inferred IRC categorizations (design model interruption, reaction, and comprehension objectives) for several notification systems. Plotted locations other than cube corners represent intermediate I, R, or C levels such as "some reaction" or "slight comprehension."* . . . 54
4.5 *The original prototype design of Scope, the application used as the focus of our case study. Scope sits in a corner of the desktop, presenting notification items as symbols within categorical quadrants. Urgency ratings correspond to centrality within radar metaphor. Scope is fully described in [126].* . . . 56
4.6 *Questionnaire and ratings used by the Scope design team, reported in [126].* . . . 57
4.7 *Questionnaire designed based on IRC claim categorization (alarm, ambient, or supertask), with ratings obtained in our user study. Apparent from the mean claim ratings (3.27, 4.03, and 3.17 respectively), the Scope facilitated ambient goals best and was most lacking in support for simultaneous (supertask) goals.* . . . 58
4.8 *Case study summary. From the previous questionnaire and our IRC-based version, we extracted information design claims and then mapped them (using arrows) to the redesign strategy actually selected and reported in [126]. Note that an (X) on an arrow denotes inconsistency between identified claim and redesign action. Clearly, the IRC-based questionnaire supported the actual redesign strategy decisions better.* . . . 60

5.1 *Summary of the four groups (one professional and three novice) from which design rationale of designed notification systems was analyzed. The results from the Expedia contestants and the CNN Notification designers were most concerning, as the design rationale consistently lacked focus on key psychological effects of notification.* . . . 71
5.2 *Three alternate approaches to generating a design model IRC (categorical, additive, or guideline based) that were initially considered. All three approaches were discarded after prototype methods failed to suggest consistent selection of choices, feasible scalability for an entire design space, and for other reasons discussed in Section 5.3.1. Instead, the IRCspec tool was developed to implement the concepts depicted in Figure 5.3 and the algorithm described in Appendix A.2.* . . . 74

5.3 *Flow chart of the IRCspec system.* . . . 76
5.4 *Distributions of the design model parameters samples collected using the IRCspec method and the manual method. Polynomial trendlines (4th order) provide a sense of the distribution shape.* . . . 83

6.1 *Equation for the Interruption (I) critical parameter.* . . . 96
6.2 *Equation for the Reaction (R) critical parameter.* . . . 98
6.3 *Equation for the Comprehension (C) critical parameter.* . . . 100
6.4 *Unsimplified version of the C equation* . . . 100
6.5 *Notification system interfaces studied in the case study usability evaluation. The three prototypes are redesigns of the original Scope interface.* . . . 107
6.6 *Design model (DM) and user's model (UM) assessments for Prototype A, evaluated in the case study.* . . . 109
6.7 *Design model (DM) and user's model (UM) assessments for Prototype B, evaluated in the case study.* . . . 111
6.8 *Design model (DM) and user's model (UM) assessments for Prototype C, evaluated in the case study.* . . . 111
6.9 *Notification system interfaces developed to keep users informed about Expedia.com airplane ticket prices, as studied in the case study usability evaluation.* . . . 113

7.1 *The process we followed to establish claim content, describing design knowledge of a notification system artifact.* . . . 126
7.2 *An example of the claim record development, emerging in steps that followed the claim creation process: a) describing design knowledge of a notification system artifact; b) sample claim description, scenario, upsides, downsides; c) claim fields relating to consideration of notification effects, pertaining to the Notification Collage example.* . . . 128
7.3 *Screenshot of the faceted information retrieval interface for the notification system claims library.* . . . 130
7.4 *Example of the help screen features added to the faceted search page for the second user study.* . . . 134
7.5 *Design model and user's model plots of a notification system, revealing the reengineering approach necessary. Reengineering approach trajectories may be reusable, revealing common design problems. Design models can be associated with development resources that are specifically adapted for the combination of user goals. In this example, the design model is an ambient interface (a notification system sub-class, characterized by IRC region), so a design catalog would return claims and other resources that have been shown to be successful for that blend of user goals.* . . . 140
7.6 *General architecture of LINK-UP's design-support modules. The light grey region in the center depicts Norman's conceptual models (Norman, 1986), which are extended through our work. Numbers refer to steps though the process, and are referenced and explained in the sections that follow.* . . . 143

7.7 *The high-level activities supported by LINK-UP's Requirments Analysis Module (external HTA tool is not currently integrated.* . . . 145
7.8 *An example of a stage of action task decomposition for an Alarm task, such as "Respond to Important Alarm."* . . . 145
7.9 *The three high-level activities supported in the current implementation of LINK-UP's Participatory Negotiation Module.* . . . 148
7.10 *The high-level activities composing LINK-UP's Analytic Evaluation Module.* . . . 149
7.11 *Claim relationships, described in the bulleted points above, to appear in [129].* . . . 155

A.1 *Example design model IRC calculation process.* . . . 193
A.2 *Results of the manual walkthrough.* . . . 208
A.3 *System choices for Problem 1.* . . . 209
A.4 *System choices for Problem 2.* . . . 210
A.5 *System choices for Problem 3.* . . . 211
A.6 *System choices for Problem 4.* . . . 212
A.7 *Instructions provided to participants who did not use the IRCspec tool.* . . . 213
A.8 *Response form used by participants using the IRCspec tool. On this form, the participants indicated their choice for the system that would the best starting point for thinking about the design problem.* . . . 214
A.9 *Response form used by participants (not using the IRCspec tool) to record IRC rating assessments.* . . . 215
A.10 *Comparison of IRC-rating consistency between two experts. Both provided manual IRC estimations before using the IRCspec tool. Neither expert had used the tool before the experiment. Parameters calculated using IRCspec were not seen by either expert during the experiment.* . . . 216
A.11 *Interrater reliability of IRC-rating consistency between two experts, calculated using Kendall's tau.* . . . 217
A.12 *Raw IRC-ratings provided by the participants using each of the two methods (manual and IRCspec). Cell shading shows the general difference between ratings and the expert score (calculated with the same method).* . . . 218
A.13 *Absolute differences between participants IRC ratings and expert ratings (calculated with the same method). Differences greater than 0.3 are highlighted.* . . . 219
A.14 *Rating ranges for each of the four design problems. Expert Benchmark scores are shown with a +/- 0.1 range, and participant ratings means are shown with +/- one standard deviation. From this depiction, it can be inferred that most participants using the IRCspec tool would hit within the expert range, except in four cases (75%): C-value for Problem A, I-value and R-value for Problem C, and the C-value for Problem D. On the contrary, most participants using the Manual Method would only hit with the expert range in two cases (12.5%): C-value for Problem B and I-value for Problem C.* . . . 220
A.15 *Interrater reliability of raw data.* . . . 221
A.16 *Interrater reliability of difference scores between participants and Expert 1.* . . . 222

B.1 *Summarized justification and sample results for the I equation.* . . . 227
B.2 *Summarized justification and sample results for the R equation, continued on the following page.* . . . 228
B.3 *Summarized justification and sample results for the C equation.* . . . 230
B.4 *Script used for the analytical IRCresults procedure, continued on the following pages.* . . . 232
B.5 *Form used by the expert participants in the experiment, continued on the next page.* . . . 235
B.6 *Raw data (IRC ratings), team means, and differences between individual scores and team means and the overall mean.* . . . 238
B.7 *Interrater reliabilities, showing tendency of an individual to agree with their team's mean rating. The second test was conducted without Team 3's results (this team was an outlier in terms of internal consistency).* . . . 239
B.8 *Raw data (IRC ratings) for Interface B, rating group means, and differences between individual ratings and team means and the overall mean of all three interfaces.* . . . 240
B.9 *Raw data (IRC ratings) for Interface A, rating group means, and differences between individual ratings and team means and the overall mean of all three interfaces.* . . . 241
B.10 *Raw data (IRC ratings) for Interface C, rating group means, and differences between individual ratings and team means and the overall mean of all three interfaces.* . . . 242
B.11 *Specification for the Expedia notification systems.* . . . 244
B.12 *Data provided to Expedia notification system designers, consisting of times at which ticket prices for eight different flights change.* . . . 247
B.13 *Experiment setup criteria for the LAX flights, as well as resulting signal definitions, comprehension and projection questions.* . . . 248
B.14 *Experiment setup criteria for the LGB flights, as well as resulting signal definitions, comprehension and projection questions.* . . . 249
B.15 *Raw data and user's model IRC values obtained from the LAX round.* . . . 250

C.1 *Modified claim format, where the first 14 fields correspond largely to Sutcliffe (2002).* . . . 252
C.2 *Prototype of the claim record display screen (continued on next page).* . . . 253
C.3 *Screenshot of the claim adding mechanism within the claims library. Four screens (like this one) allow the user to add all information about the claim, such as associated primary and notification tasks, IRC values, and multimedia objects that depict the artifact.* . . . 255
C.4 *When an owner of a claim opts to edit the claim already stored within the claims library, this screen appears. On this page, the user selects the portion of the claim record he want to change. The screen shown on the following page allows the editing operation to occur.* . . . 256
C.5 *Rating page for administrators to subjectively evaluate the quality of claims. All unrated claims appear in the left frame. The user is able to access information related to the other three rating factors (frequency of claim use in projects, average user rating, and claim author's experience level) with the link next to "Current Rating". Fields associated with the theory and artifact's multimedia components are accessible with respective links.* . . . 258

C.6 *Screen within the Requirement Analysis Module, allowing a user to specify the design model IRC parameters (using IRCspec), as well as basic tasks include in the problem scenario. At this point in the interaction sequence, the user would have already created a project and entered problem scenarios.* . . . 260
C.7 *After using the IRCspec tool to obtain design model IRC values for the a problem scenario, the designer is presented with recommendations for a generic task model. In the fully developed modules, many generic task models would be available, based on different combinations of IRC values and information processing strategies (i.e., top-down or bottom-up). Other information obtained from designers in the IRCspec interaction, as well as through the selection of basic tasks (generic and generalized) can also help to provide template recommendations.* . . . 261
C.8 *In the step shown here, the designer selects the specific generic tasks that occur within each stage of action, as well as a pre-defined sub-task model (not yet implemeted) that indicates likely IRC changes within each stage of action.* . . . 262
C.9 *In this final step, the designer selects or creates new problem claims that elaborate on the sub-tasks within each stage of action (defined in the previous step). Claim possibilities for each sub-task can be recommended, since the IRC characteristics have been defined, as well as the generic tasks. This process results in a problems claim list according to stage of action.* . . . 263
C.10 *As one of the initial states within the Analytic Module, this begins the process of the designer creating a record of their system image. From here, the designer can later access the evaluation results, once they have been completed by the expert evaluator.* . . . 264
C.11 *The designer begins the system image creation process by entering administrative data about their design and associating the to-be-created system image with an existing project file.* . . . 265
C.12 *After associating the system image with an existing problem file, the problem claim set (established in the Requirements Analysis Module) is loaded. Here, the designer enters new or selects existing design claims that address each problem claim. This process is continued for each problem claim.* . . . 266
C.13 *Once all of the problem claims and design claims have been entered into the system and linked, the designer is able to modify or validate their design model IRC (using IRCspec), enter interaction scenarios describe how their prototype works (or is intended to work), and upload descriptive screenshots, movies, or other material.* . . . 267
C.14 *With the system image established and made ready for evaluation, an expert evaluator is now able to enter the system, access the system image, and inspect design rationale and descriptive material. Here, the evaluator is able to step through the problem-design claim pairs.* . . . 268
C.15 *For each problem-design claim pair in the claim set, the evaluator can indicate whether the claims represent a design or usability flaw (commenting as appropriate) and inspect the associated attachments.* . . . 269

C.16 *After inspecting the full claim set, interaction scenarios, and supporting materials (screenshots, etc), the evaluator completes the analytic IRCresults questions to obtain an analytic model IRC. These values and associated comments to clarify the rating become part of the evaluation record.* . . . 270
C.17 *Here, the designer is provided with a report of the evaluator findings. These findings include a comparison of the design model and analytic model ("Estimated") IRCs, as well as a summary of the flaws found with the claim set.* . . . 271
C.18 *After executing the empirical testing and uploading the participant's output files, the Empirical Testing Module parses the files and separates IRC data according to each claim, as defined by the designer's original specification.* . . . 272
C.19 *After the empirical testing results are viewed by the designer, they can be permanently saved to the associated claim records. The system appends the empirically determined IRCs to the claim record in a way that preserves the testing context (i.e., the "Widget_NS.exe" would be a link to the empirical test definition, reusable test script, and test platform.)* . . . 273
C.20 *Responses related to the Empirical Testing Module from the seven designers that participated in the pilot testing. Means and standard deviations are shown on a scale from "strongly agree" (SA) to "strongly disagree" (SD).* . . . 274
C.21 *Responses from the seven designers that participated in the pilot testing for both the Analytic Module and the Empirical Testing Module.* . . . 275

List of Tables

3.1 *Significant attribute orderings, by cognitive tasks ($p < .05$). "Low Degradation" refers to trials where primary task degradation was less than seventeen percent, while "High Degradation" was seventeen percent or greater.* 29

4.1 *Taxonomy of notification systems relating design parameters to implementation options or considerations. We suspect that most notifications will be designed and used in a dual-task situation with some expectation of performance loss for each task.* 39
4.2 *Attention benefits and costs. Notification system users expect to gain benefits associated with fulfillment of user goals (left side) by sacrificing attention from other tasks. Costs can be exacerbated by factors of the current situation (right side).* 42
4.3 *Descriptive names, IRC categorizations, and example scenarios for the notification systems design space, forming "scenario families" for notification systems.* 48

6.1 *Proposed critical parameter components.* 94
6.2 *Concrete terms used in the I equation, with usability evaluation assessment techniques for each.* 98
6.3 *Concrete terms used in the R equation, with usability evaluation assessment techniques for each.* 100
6.4 *Concrete terms used in the C equation, with usability evaluation assessment techniques for each.* 102

A.1 *Design rationale of experienced, professional developers and researchers, from published papers (continued on next page).* 187
A.2 *Design rationale from student design projects (continued on next two pages).* 189

B.1 *Relationship between questions used in the IRCspec tool and IRC equation variables, showing values used for answer scores. Table continues on the next page.* 224
B.2 *Concrete terms used in the I, R, and C equations, and usability evaluation assessment techniques for each.* 226

Chapter 1

INTRODUCTION

As people everywhere become increasingly more insistent on integrating additional computing tasks with routine and critical daily activities–a behavior fueled by fervent demand for pervasive and ubiquitous information–a gap within HCI research grows. Certainly, much progress has been made toward understanding and refining typical desktop interfaces used during extended periods of concentrated attention with orderly, predictable task action flow. However, different usage situations, expectations, and error consequences govern the growing breed of applications and devices being introduced to support multitasking information demands. Referred to as *notification systems*, these interfaces are generally desired as a means to access valued information in an efficient and effective manner without introducing unwanted interruption to a primary task, and can be found in many implementation forms and on a variety of platforms. Perhaps classic desktop systems are the most readily identifiable: instant messengers, status programs, and news and stock tickers. However, other familiar examples such as Weiser's dangling string representation of network traffic [132], in-vehicle information systems, ambient media, and multi-monitor displays hint at the range of potential notification systems once we consider off-the-desktop information delivery mechanisms. While this range of solutions has skyrocketed, our ability to scientifically recognize, pattern, and improve success within these systems has not kept pace.

Even though use of these systems has become widespread in recent years, there are surprisingly few efforts within HCI literature that effectively evaluate usability of the information and interaction design for notification systems. For example, while some notification systems support collaborative activities and are studied from a CSCW perspective, disparate agendas lead to inconsistent definitions of successful design, inhibiting cross-initiative influence. In other cases, notification systems are ubiquitous computing devices that a have strong multitasking element. From the ubicomp perspective, HCI researchers are interested in issues like context-awareness and privacy, without focusing on the narrower questions of notification delivery. As one of the two important research challenges asserted by Abowd and Mynatt for the ubiquitous computing field, they motivate the imperative for assessing progress toward real human needs with quantitative and qualitative evaluation methods that capture authentic context of system use: "research in ubiquitous computing will have limited impact in the HCI community until it respects the need for evaluation" [1].

As a starting point, much of the dual-task experimentation (especially cockpit design) performed within the human factors and engineering psychology fields seems highly relevant to this area of research (we provide a summary later), however, this body of science does not seem to be readily applied to notification systems design justifications. While some early studies of notification systems have captured useful guidelines and design tradeoffs and serve as initial models [41, 81, 85], few efforts have been conducted and reported to explicitly afford knowledge application and reuse, facilitate study replication and extendibility, or even proceed along a standard evaluation methodology-clearly objectives of empirical and analytical evaluation. An umbrella approach is needed, tying together knowledge and addressing challenges in notification systems design throughout the HCI community. Through the work proposed, developed, and pilot-tested in our research program, we introduce the *IRC framework* as a guiding conceptual approach [87, 88]– a unifying framework for understanding, classifying, analyzing, developing, evaluating, and discussing notification systems that will promote scientific growth and knowledge reuse.

Problem statement. The IRC framework (fully described in chapter 4) provides a modeling method for describing user notification goals and information delivery effects. Although we have received many positive reviews and feedback about our preliminary versions of the framework from other researchers and we have observed further research and educational benefits though using the framework in seminars, it still needed to be sufficiently developed to as a reliable and consistent design tool. Furthermore, studies and follow-on work had yet to be undertaken that would assess the hypothesized utility that the framework delivers during design phase processes. Our overarching research problem can be succinctly stated:

> To promote the IRC framework as a unifying framework for notification systems research and apply it to improve design discourse, critical service components and features must be added and tested to ensure consistent, reliable results and a favorable impact on the design process.

As we consider how to overcome this problem, we recognize several implied tasks and constraints. First, we want our design support system to be useful to researchers anywhere. This will impact a determination of how to add widely accessible components and features, as well as the selection of test criteria. Second, we want our framework to be compatible with dominate ideas in the HCI field, especially those used in education. We have selected Norman's cognitive engineering concepts as a foundation on which to build, adding to the initiatives already started by Carroll and Rosson, Sutcliffe, and others. Third, as we consider what aspects of the "design process" we can really contribute to from an HCI perspective, we focus on two design activities: 1) recognizing whether user requirements are met by design artifacts, and 2) storing and accessing potential reusable design knowledge.

1.1 Motivation & Vision

The research problem, implied tasks, and constaints are transformed into our research vision. By developing key functionality and services of the IRC framework, we enhance and demonstrate the utility of directly comparing a notification system's design and user's model (in the sense of Norman's conceptual models [100]) to support design benchmarking and enable design reuse. Several recent dialogues within the HCI research community provide further underlying vision. First, there is enormous potential when psychological models are applied to create macrotheories describing interactions within a mental architecture [9, 10], especially as a basis for early-phase, predictive usability evaluations. The IRC framework is motivated by this idea, but extends the concept with a literal interpretation of Norman's argument that usability engineering should be driven by mental model comparison and consideration. Second, we recognize great long-term benefit in approaches such as the systematic establishment of critical parameters based on Newman's ideas [97] and the reference task agenda argued by Whittaker et al. [133], both of which lead to cohesive, community research efforts. The basic tenets of the IRC framework are abstracted user goals and usage consequences, which serve as critical parameters that can gauge design progress according to reference tasks. Finally, Sutcliffe's notion of "claim families," which he advocates as a mechanism for reuse within a scenario-based approach [117], is also quite promising as a method for incrementally improving design guidelines and increasing efficiency of requirements engineering. Since the IRC framework should be able to assist problem, activity, information, and interaction design claim generation processes, it seems like a natural mechanism within a claims library.

With this impetus, this research delivers enhancements to the IRC framework and presents testing and analysis of its main functions. The enhancements can be described generally as services that ordinary designers (with no knowledge of the IRC framework) can avail during early phase design of a notification system, enabling consistent and accurate representation of their system objectives, usability testing results, and access to archived design knowledge. The tests and analyses reported in this dissertation show the level of confidence we can have about the IRC framework's support for these tasks and motivate directions for future work.

1.2 Goals and Objectives

Goals. The framework enhancements, testing, and foundation for future work will address critical services, features, and functions within three design processes: system description, interface evaluation, and design comparison and reuse. *System description* is intended to refer to the mechanisms that a designer has available to describe design intentions, interpretations of design requirements and specifications, situational variables, and anticipated effects on the user. Describing a system helps a designer articulate the requirements analysis, assists an evaluator with selecting an appropriate usability testing strategy, and allows an implementation team to understand the designer's vision. *Interface evaluation* is the process within usability engineering that assesses whether the design would serve its intended purpose for users. *Design comparison and reuse* allows more efficient progress within an individual design cycle or research area, since one design can be compared with another, allowing strengths and weaknesses to be identified, contrasted, and leveraged in new designs. Each of these three design cycle processes requires systems to be developed so that IRC functions can be implemented, evaluated, and used in a broader research program. Providing systems that serve as consistent and accurate automation of IRC processes, as well as conducting an analysis on each, serve as the overall research goals that will achieve the vision presented earlier.

Objectives. This research enhances the IRC framework by providing three systems: a specification system for design model IRC ratings (referred to as *IRCspec*), a conversion system that reduces usability testing results to a user's model IRC rating (or *IRCresults*), and an initialization of an IRC-compatible claims library and design environment (*LINK-UP*). IRCspec supports our goal toward system description, IRCresults facilitates interface evaluation, and LINK-UP would enable design comparison and reuse. After a description of each systems' design rationale and development process, results from testing providing insight into the potential for IRC representations and comparisons of design and user's models, as well as the ability of the IRC framework to support design benchmarking and cataloging. The processes involved in achieving these objectives also result in many other products, such as design and development of novel notification systems, collection of notification system design artifacts and claims, and creation of a highly reusable task-specific, application-generic usability evaluation tool.

1.3 Anticipated Impact

The most important contributions of this research are not in the immediate resulting individual products, but in their synthesis as research infrastructure that will be applied and extended through years of continued work. A few initial directions have been outlined, to include efforts that will investigate HCI approaches for a science of design, improve educational materials for HCI classes, and facilitate multidisciplinary participation in interface design research. As notification systems play a prominent role in a wide variety of domains, these tools can be used to support exciting branches of interface design. In the long-term, the results of this research are expected to contribute to the HCI and notification systems research communities in several ways:

- *Better notification systems*—adding utility, value, and enjoyment to the user experience through improved usability engineering, with the IRC framework allowing meaningful comparison of designer and user conceptual models.
- *Less costly production of new systems*—resulting from the IRC framework's support for pragmatic requirements analysis, interface evaluation methods that capture usability problems early and accurately in the engineering cycle, and claims catalogs that enable reuse.
- *Faster cohesion of research community findings*—benchmarking of reference tasks to allow greater generalizability, extendibility, and replicability of design performance and research.

- *Positive research program example*—providing support for methods founded on scenario-based design, critical parameter modeling, cognitive architectures, usability evaluations, and claims reuse. This approach should inject science within HCI and is thought to be applicable to other classes of systems.

1.4 Overview of the Dissertation

This section describes the organization of the document and is intended to serve as a guide for selective reading. In general, chapters 2 and 3 motivate the problem statement, chapter 4 introduces the IRC framework, chapters 5 and 6 describe design process support tools, and chapters 7 and 8 demonstrate the integration of the IRC framework with interface design and other HCI activity (i.e., research and education).

- **Chapter 2**–This is a formal literature review that will introduce a reader to the notification systems research field and demonstrate understanding of several important HCI paradigms. The review focuses on specific studies of usability evaluation efforts that were carried out to improve understanding of delivery and display notification information. In addition, we review broad movements and concepts within the HCI community that are foundational to our research. Review of other literature appears in context throughout the document.

- **Chapter 3**–This chapter, referred to as *background work*, describes early empirical testing work done by the author to investigate information design options for secondary displays (a specific type of notification system). Patterned after studies like those reviewed in the previous chapter, the efforts to produce generalizable design knowledge suggest the need for a more structured understanding of the notification systems research leading to the work presented in the next chapter. The conclusions from this empirical research are later revisited, as an example of how the IRC framework can modify our approach to archiving usability evaluation results.

- **Chapter 4**–As *preliminary work* that is extended throughout the remainder of the document, this chapter presents the details of the IRC framework, which have been published in HCI journals. In addition, the ideas presented here have been discussed at the workshop for design and evaluation of notification systems at UbiComp 2002 and CHI 2003 and include response to critical feedback received from external reviewers. The proposed research directly extends this conceptual work to a point that it can be applied in a design cycle. Understanding the basic ideas of the IRC framework is essential for understanding the motivation, processes, and significance of the design support tools, the evaluation methodology used to assess their accuracy and consistency, and the ideas that integrate the tools with design process and a long-term research agenda.

- **Chapter 5**–To facilitate a designer's estimation of design model IRC ratings during the initial stages of a notification system development effort, this chapter describes the design and testing of a software tool–IRCspec. Tool development decisions were based on our requirements analysis effort (reported in this chapter) that explored professional and novice designers' tendencies to include consideration of critical parameters in their design rationale. After discussing the specific development decisions, lab-based testing results for the tool are presented. Throughout the description of IRCspec tool and reflection on implied challenges to using critical parameters in a design process, we note specific activity claims as a summary of IRCspec's design rationale.

- **Chapter 6**–While IRCspec helps designers determine design model IRC ratings, a similar process/tool is required for abstracting usability evaluation results into a user's model IRC rating. This is the focus of chapter 6. The analysis begins by addressing the challenges with using critical parameters, as previously noted. This inspires an elaboration of the critical parameter concept, which preserves the intention of a generic concept but explicitly adds a relation to specific terms. The idea is demonstrated through the development of equations for the IRC parameters, which are operationalized with usability evaluation tools (implementations of the IRCresults relations). Two case studies are presented to demonstrate the use of these tools–one that employed an analytic evaluation technique and another that used an empirical approach. The analytic evaluation case includes a study of the tool's consistency and accuracy. Activity claims are also noted throughout the discussion of the IRCresults tools.

- **Chapter 7**–In this chapter, the focus shifts to an examination of how the IRC framework and its associated tools can be used throughout a design process to achieve goals of design comparison and reuse. As a specific approach, we explore tensions inherent with using the IRC framework as an indexing mechanism for a claims library. An initial implementation of a claims library for notification systems is described, and preliminary usability evaluation results are presented. To mitigate the difficulties observed by participants, we broaden the usage concept of the claims library into an integrated design environment, referred to as LINK-UP. Activity design, IRC integration, and preliminary user tests are reported for four design-support modules of LINK-UP. This portion of the document lays significant groundwork for the future work described in the next chapter.

- **Chapter 8**–In order to present a more coherent "big picture" view of this research, the final chapter includes an integrative case study that examines the use of the IRC framework, its tools, and the LINK-UP system in a design process. The case study begins to build evidence that this new research infrastructure can improve the design of notification systems and assist HCI research and education efforts. Broad conclusions of the work thus far are noted. With the view that well-defined future research directions are the most important products of a dissertation, the chapter develops four specific avenues for further work.

Chapter 2

RELATED WORK

This chapter presents a review of literature that is related to the research direction extended by the dissertation. To place these efforts within context with other computer science research, we begin by introducing the field of human-computer interaction (HCI), the design-science discipline that deals with improving a user's experience with software and other computer-mediated systems (section 2.1). We then introduce a specific type of interface—notification systems, the more focused concern of our efforts. To illustrate the wide range of these interface implementations and their potential effects on users, we discuss many examples of systems that appear in literature and are commonly used (section 2.2). Next, we review the various methods and approaches that notification system researchers have employed in recent years to improve the usability of these systems (section 2.3). Since the specific concern of the our research is toward enhancing research cohesion and extendibility of design efforts, we return to the broader HCI and software engineering fields to review other techniques, concepts, and arguments related to usability engineering, the process of research, and software and knowledge reuse (section 2.4).

2.1 Human-Computer Interaction and Interface Design

Central to the goal of human-computer interaction research is mediating human cognitive and perceptual capabilities and preferences within system constraints [100]. This area of research integrates many other disciplines for the express purpose of improving interfaces for computing systems. Certainly, this activity falls within system lifecycle development and can be most closely related to software engineering. However, the important issues driving this area of research often seem to involve the necessity to recompense human limitations and leverage characteristics of human behavior, so a human factors or psychology lens is often employed. This cross-disciplinary approach represents a pursuit for design guidance that allows programmers to create computer interfaces that proffer insight rather than impose information glut.

Within the study of human-computer interaction, an important topic is information design—how to physically represent data and information in a manner that best supports user processing goals. Many different influential approaches exist, especially collections of guidelines that have resulted from transfer of graph design methods and experimentation in statistics research, as well as experimentation in the cognitive psychology field. One prominent example is work done in *attribute encoding*—representing data with visual primitives such as color, shape, and position. Cleveland and McGill provide a well-known ordering of attributes most suitable for graph design, based on psychophysical theory and experimentation [40]. They recognize visual data as elementary perceptual tasks, which can be described as attributes of graphs and have been influential for interface design. Since certain attributes convey information better than others, these attributes are orderable to form design guidance based on psychophysical theory and experimentation. Tufte is also a well known contributor of information design guidelines, addressing topics such as displays for decision-making, visual parallelism [124], layering and use of color [123], graphical element aesthetics, and optimizing data-ink relationship [125]. The study of *information visualization* adapts these types of techniques and develops new approaches (such as the Visual-Information Seeking Mantra: overview first, zoom and filter, then details on demand; discussed in [112]) for making sense of complex information.

While graphical design principles form a large body of literature for information design of objects in a users focal attention, the interfaces within our concern (notification systems) are usually used outside focal attention as *peripheral displays*. Research on these systems has only begun to emerge as an area of interest within HCI, but work within the human factors and experimental psychology fields has addressed similar issues for years. Perhaps the most comprehensive review can be found in Wickens and Hollands discussion of the *dual-task situation*, where they review primary task performance degradation in terms of resource allocation to secondary tasks and adaptation consequences for excessive workload [135]. Much more information about this specific analysis can be found in Chapter 4, and other related contributions are discussed in the sections that follow.

Before continuing our discussion of information design contributions and evaluation approaches that are influential to notification systems design, we will first provide a more thorough introduction to this particular class of systems.

2.2 Notification Systems in Literature and Use

There is certainly enormous research potential for understanding how to communicate constantly changing information to interested persons at the ideal time. This problem is compounded when information is intended for a user occupied with other tasks, since the human-attention system becomes a critical factor. Traditional HCI research provides theories and guidelines for information and interface design, but falls short on many levels when applied to these particular problems.

Consideration of notification systems could be constrained to desktop interface elements, such as stock tickers, instant messaging tools, system load monitors or alerts, and the like. These types of displays share the common design goal of providing the user with information awareness without requiring excessive attention. Such interfaces should be designed specifically to minimize the cost of distraction to other tasks and to maximize the utility gained from the information being displayed. This may require special consideration of screen space, information encoding, and other interface design choices. However, people's notification needs extend far beyond the computer desktop, and so should a definition of notification systems. We think of *notification systems* as interfaces that are mostly used in divided-attention (or dual-task) situations as the lesser portion of the user's attention (as a secondary task) in a given period of time.

In recent years, the research community's efforts in facilitating use of multiple, simultaneous information sources are demonstrated by many innovative interface design approaches.

Example desktop applications. Several efforts can be characterized by their attempt to deliver information of interest with small desktop applications, specifically designed to provide glanceable awareness without disturbing other tasks or becoming annoying. Common examples of these types of applications include news or stock tickers, system load monitors, Internet browser add-on (such as ESPN BottomLine [1], which provides a display for sports scores), and taskbar icons. Several interesting applications and notification approaches have recently appeared in HCI literature, including two systems introduced by Microsoft Research—the Scope and Sideshow, and a third system referred to as Irwin.

The designers of Sideshow present a collection of small display elements in a vertical strip that is analogous to the Windows taskbar, promoting this notification system as a peripheral interface for keeping people aware of important, dynamic information without causing too much distraction [19]. Sideshow is designed upon the paradigm of awareness facilitation, which is a solution to the drawbacks involved with information polling and automated prediction of alert utility. Sideshow allows users to add and configure tickets, which visually summarize information states and allow immediate access of details.

According to van Dantzich et al., Scope was designed with similar goals of providing glanceable awareness without straining cognitive resources and causing task interruption. The system employs a single display that is an "unobtrusive" radial visualization of urgency scores for various items of information (such as items in an email inbox or on a task list), based on learned or specified user prioritizations. Features are similar to Sideshow's—easy access to details, some microvisualization, calm updating, and customization options, but the Scope also serves an example of automated classification and presentation of information items [126]. This interface (pictured in Figure 4.5) is discussed in much more detail in section 4.3.2.

Yet another example of a desktop notification system interface is Irwin [83, 87]—a small, omnipresent tool that assists users in maintaining awareness about Internet resources such as email folders, Usenet newsgroups, web pages, and weather data. Information is gathered from several sources and displayed on a central visualization; various icons, colors, and auditory cues keep its user updated. With each of these three applications, users are able to receive continuous updates within their desktop screen space about information of interest while they work on other tasks.

As an alternative to dedicating constrained screen space to tickering displays and other no-

[1] http://espn.go.com/bottomline/

tification tools, Harrison et al. argue that transparent user interface elements, as layered, space multiplexing technique, can provide awareness of other information and enhanced context while minimally disrupting focused attention on standard interface objects [57]. Other desktop applications, usually intended to be used with other tasks, seem to take an approach of not being concerned with preventing distraction. Instead, they proactively provide prompts that are intended to guide or enhance activities. Certainly, Microsoft's Office Assistant (Clippit) and Rhodes and Maes' Remembrance Agent [104] are examples of these types of applications. Other examples include instant messengers, email and schedule alerting features, notifications within groupware tools, and music visualizations.

Notifi cation beyond the desktop. Other innovative work has demonstrated feasibility and utility of presenting notification information within a user's environment, although there are many different approaches here as well. Large screen displays are used in both MacIntyre's Kimera augmented office environment [78] and efforts like Informative art [103], but there are fundamental differences in the objective amount of user attention necessary to extract information and gain meaning. Kimera's wall displays seek to provide quickly understood background awareness cues that complement the flow and context of work, while Informative art provides a hidden representation data that is enjoyed during moments of deeper reflection. Techniques for subtly altering elements of the user's environment to convey information for background processing was demonstrated in the ambientROOM and elsewhere with projections of water ripples, natural soundscapes, spinning pinwheels, patterns of light patches, and the Information Percolator's air bubbles [69, 42, 60]. Other work has described how physical widgets (called phidgets) were produced to display information states with curious physical objects, such as an artificial flower arrangement or Phidget eyes [55].

Although many of these examples are designed to enhance user efforts on desktop platforms, in classrooms, and in office environments, similar research interest (and HCI expertise) often extends to cover more ubiquitous displays, such as vehicle and wearable navigation/information systems, heads-up displays (HUDs), and augmented reality applications. Collaboration tracking and groupware systems also tend to have multitasking design components, where information of interest is presented in a divided-attention situation.

One of our initial concerns with the notification systems research area is that the definition seems boundless, preventing focus on specific design challenges for which information design solutions can be collected and compared. In the next section, as we consider some of the research that has been conducted for these and similar systems, an organizing theme begins to emerge.

2.3 Evaluating Usability of Notification Systems

Understanding the impact of information design differences in notification systems has provided direction for several HCI research efforts, including many of our own recent investigations [37, 120, 114, 34]. These research efforts have the common objective of determining specific and comparative effects of variations in information encoding. There seem to be at least two general approaches to analyzing factors of notification presentation: preserving primary task attention demand and maximizing the utility delivered during brief attention reallocation.

2.3.1 Preserving attention demand

Some efforts within the community have focused on achieving effective attention allocation within a dual-task system by reducing distraction of notifications as much as possible. Much of the prior work on distraction in notification systems considers secondary displays for in-vehicle information systems, where distraction from the primary vehicle control task can be harmful or fatal. Guidelines established in these areas suggest defining limited numbers and types of interactions with the displays, restricting the amount that displays change, and limiting the time that a display is present [8, 54, 76, 122, 111]. In most desktop computer usage situations, when the consequences of distraction are not life-threatening, annoyance threshold seems to determine the amount of distraction that is acceptable, although research suggests that performance on an interrupted task will suffer for longer than simply the time required to perform the secondary task [7]. Perhaps these types of guidelines are most suitable for situations that require a notification system to not intrusively disrupt user attention devoted to a main task. However, as ubiquitous notification displays and devices increase in popularity and are coupled with more attention-intensive primary tasks, understanding how to satisfy this design objective becomes increasingly more important.

Other situations explicitly call for notification prompted task-switching or provide some tolerance in allowing interruption to the primary task in order to accommodate acquisition of secondary information. In these cases, there is usually still some value associated with minimizing interruption before unacceptable primary task performance degradation occurs [135]. To address these situations, other research approaches have sought to optimize selection of attention demands by considering associated cost of user interruption and appropriately tailoring notification presentation. Horvitz's models and Bayesian inference procedures present some hope for this design objective, an imperative driven by his belief that human attention is the most valuable commodity in HCI [66, 64]. These models are designed to improve notification utility by considering cost of user interruption and introducing notification presentation appropriately—a strategy employed by Microsoft's Notification Platform [67].

To support this type of emerging notification adaptivity, we must be as certain as possible about comparative interruption properties of information design attributes so that they can be properly mapped to interruption levels. However, selection of information design for a notification system that is driven by inferred suitability of interruption will likely have impacts on the objectives and affect overall system utility. An approach like this is useful for filtering information to be presented to a user interested in receiving valuable notifications, such as the receipt of urgent email or a reminder for an important meeting.

As yet another approach to minimizing the attention drawn away from the primary task, recent work by McFarlane presents additional applicable background for understanding aspects interruption for attention management through a notification system interface. He provides results of empirical studies that evaluate four design implementations to coordinate interruptions (immediate, negotiated, mediated, or scheduled) [92]. The tentative guidelines he established, which are particularly useful for supporting a user-initiated interruption design, exhibit design goal tradeoffs among the coordination methods. Negotiation-based interruption coordination appears to be best for many cases. Additionally, he introduces a taxonomy established through literature consolidation describing eight major dimensions of interruptions [91]. Likewise, Maglio and Campbell investigate peripheral presentation options that will avoid unnecessary interruption, preserve primary task attention, and accomplish notification delivery [80].

2.3.2 Maximizing attention utility

Information design evaluations for notification systems have also included studies that seem to accept some given amount of attention reallocation and focus on using that attention for as much as possible. To this end, some researchers investigate how notification delivery can be optimized to provide users as much utility as possible during whatever interruption is available. Since many users wish to stay informed about values of information of interest with minimal interruption, several studies have investigated how notifications can be accurately detected and responded to using preattentive processing, considering how information can be assimilated and understood rapidly with different colors, shapes, and motion [58, 59, 13]. Specifically, the research of Bartram considers the effectiveness in using motion cues to enable signal detection, identification, and reaction. This work examined the speed and accuracy with which motion cues can draw a key-pressing reaction, relative to other visual attributes like color and shape. The findings showed that for this purpose, motion cues outperform static representations in displays in the periphery of the screen. Earlier work examined moving and changing text as a method for presenting information displays, observing the perceptibility and readability of rapid serial visual presentations (RSVPs) of letters, strings, and words [45].

Rather than optimizing displays for quick glances, another approach has been to increase utility with information design options and allow deeper understanding and memorability [72]. For example, Cutrell et al. investigated impacts of messaging on primary task memory and performance [41]. Likewise, Cadiz's Awareness Monitor system was design to semantically and functionally deliver notification utility, using a series of effects (including tickers) to address both immediate interpretation and awareness gain over time [21]. Maglio and Campbell performed a series of dual-task experiments to examine the tradeoffs in displaying information using animated textual displays [80]. Participants performed a series of primary tasks where they were asked to edit a document, during which they were tested on how well they remembered notification information. While most of their findings focused on differences in distraction to the primary task based on scrolling direction and additional cue presentation, in a similar experiment McCrickard et al found other differences in the effects of peripheral animation for supporting quick and accurate monitoring or long-term awareness gains [85, 84].

While many of these efforts seem promising, they are infrequently applied to actual notification design, perhaps because few (if any) large collections of notification design guidelines exist in convenient form. Unfortunately, even published accounts of notification systems (such as those described earlier) are usually limited to implementation reports and only provide cursory hints at effective information design strategies. Fortunately, we are able to return to the broader field of HCI to review other ideas that can improve notification systems design.

2.4 Improving Usability of Notification Systems

As we consider how the notification systems research field and general design efforts can be systematically improved, we draw from many HCI ideas. The ideas are reviewed here, and revisited later in the context of the proposed research. Three general categories of HCI knowledge and methods can be potentially beneficial to notification system design: usability engineering approaches, arguments and methods for enhanced research organization, and the movement for software and design reuse.

2.4.1 Usability engineering approaches

Usability engineering is the early activity within a software design process where usage goals are planned, prototyped, and tested with special concepts, tools, and techniques throughout the design cycle. By considering the outcomes of system use that are desired by various stakeholders, and weighing them with situational factors and tradeoffs between design options, usability engineers determine what functionality a system should include and how that functionality should be presented [108]. As this is a rich field within HCI, we focus our review on process and approaches to interface design, rather than prototyping, evaluation, documentation, or other concerns.

There are a wide variety of different methodologies for interface design, such as Denning and Dargan's action-centered design [44], Watzman's Information Design Process [130], and many others. We focus our review to a suite of design concepts that are compatible with later concepts presented in this section that serve as background for our research. In recent years, much work has been done to establish and refine theories that apply to interface usability and to convert them into tools that can aid the design process. As an example, activity theory can guide understanding of the structure, development, and context of human activity, but the Activity Checklist focuses application of this theory for the concerns of interface design and evaluation [73]. By understanding underlying theory and finding new ways to make it useful for designers, we believe that general notification systems design can be improved.

Norman's conceptual models. Certainly, one key theory in interface design literature is Norman's theory of action [100]. He presents interface design as an enterprise that assists users in the accomplishment of their tasks. Since tasks are composed of psychological goals and intentions and are accomplished with control mechanisms that physically manipulate system states, he recognizes two different expressions of a task (physical and psychological) that must be resolved within a human-computer interaction system. The theory of action describes the cyclical evaluation and execution of tasks: users cross the *Gulf of Execution* to translate their task goals and intentions to action plans and execution steps within a physical system and then return to a psychological state of goal assessment and task continuation by crossing the *Gulf of Evaluation* by perceiving, interpreting, and making sense of the information displayed on the physical system.

In his argument for a cognitive engineering approach to interface design and to develop his theory of action, Norman established the idea that a usage experience is governed by the consistency of two conceptual models mediated with system input mechanisms and output display: the *design model* held by the designer and the *user's model* that is based on the user's understanding of the system [100]. To facilitate a user's evaluation and execution of tasks, designers must develop conceptual models as they would develop the scaffolding of a bridge. Several factors contribute to each of these conceptual models. The design model should be inspired by a requirements analysis that includes consideration of a user's background, situational context, and task-oriented goals. This model expresses the designer's understanding of user needs and is a representation of the intended functionality for the system. The user's model is formed by the user's understanding of the *system image*, the physical system and its documentation. The key idea we continue with is that Norman's view of the role of an interface designer is to develop the system image so that the user's model and design model are compatible.

Scenario-based design. Scenario-based design is an approach to interface development, providing an inquiry method to help designers reason about elements of a usage situation and receive participatory feedback from stakeholders [108]. Through the development and sharing of *scenarios*, or narrative descriptions of users solving problems with designed systems, designers are able to create the scaffolding across Norman's Gulfs of Execution and Evaluation [100]–and develop systems with design-user's model compatibility. During the design process, many compromises are made, but *claims* articulate the positive and negative effects (tradeoffs) of a feature on a user for accomplishing a task. Claims can address a wide variety of situational and interface aspects that affect the compatibility between the design-user's models, such as user satisfaction and feeling of reward, color and object layout, and strength of affordances. To test interface usability, developers can focus on validating key claims associated with essential supported tasks. We believe that this approach can improve design and understanding of notification systems, especially since it can help designers reason about key tasks and make claims that describe design consequences.

Adaptive presentation. Design approaches like scenario-based design usually are intended to assist a usability engineer in identifying optimal features and information presentation choices. Following a typical approach, the designer would specify these interface decisions prior to implementation, although perhaps some customization options could alter the interface during actual use. A very different paradigm exists with approaches that allow adaptive presentation displays, or interfaces that automatically alter their characteristics based on usage conditions.

Interface design in this area can be divided into two broad categories: rule-based adaptive displays and software agents. Rule-based adaptive display systems regard actual information presentation as a variable that is only defined prior to rendering. Presentation can vary according to changes in information type, extraction goals, user characteristics, or other system factors. Likewise, information content (level of detail) can by dynamically adapted. Software agents are similar, since they use artificial intelligence techniques to dynamically invoke actions and make interface decisions. Examples of actions include initiating alerts or modifying information presentation in a way designed to distract the user from their task. Review of research in both areas follows.

Mackinlay's APT (A Presentation Tool) was a first coherent effort at automated graphical design, specifically for 2D relational data. Presentation is adapted according to an information sets compositional algebra result describing relation of information semantic and syntactic properties to encoding choice expressiveness and effectiveness. While this was an important paradigm, Mackinlay acknowledges that as presentation tools are designed to handle broader ranges of information types, rules for encoding effectiveness become difficult to establish and agree upon [79]. Casner describes several other systems that automatically decide graphic formats, and tries his hand at creating such a system, seeking to maximize cognitive utility by including consideration of information-processing tasks as well, but again only demonstrating a very narrow automated presentation domain [30]. Sutcliffe and Faraday successfully extend the goal of automated presentation to multimedia information, according to a predefined Entity-Relationship diagram linking information formats and types to dialogue acts [119].

In the area of software agent research, Horvitz uses his LookOut tool, an automated scheduling service for Microsoft Outlook, to consider principles for what he calls mixed-initiative user interfaces–an intelligent agent attempts to guess user goals or problems and collaboratively offer solutions [64]. Such a service is motivated as a valued-added extension to direct manipulation. He uses dynamic analysis of a user's expected utility resulting from possible agent action to determine required threshold probability of a user having a particular goal or problem, thus supposedly balancing costs and benefits of interrupting a user. While this decision making process is only effective as the data that provides the underlying basis for establishing expected utility and threshold probability, Horvitz's enduring contribution to software agent research should be the twelve critical factors for effective integration of automated services, which he defined in his work. Brown et al. take a similar approach to improving agent prediction of user intent, focusing on using utility theory and functions to that drive a separate dynamic "adaptation agent's" correction of user models [18]—an interesting approach which opens questions about how to select proper adaptation agents. The critical theme for agent use seems be the fine lines between facilitating helpfulness or irritation, promoting reliability or over-dependence, and being well-tuned to a user or just another privacy concern.

2.4.2 Focusing research efforts

As a relatively new field of research, HCI and the notification systems research community may suffer from a general lack of research organization. That is, there seems to be little evidence that HCI research or conclusions from usability engineering practice plays a driving force in spurring innovation of interactive systems. This argument has been developed by Carroll and others to motivate the need for improved capturing of design rationale. First, we review Carroll's task-artifact framework as a proposal that would allow development of design rationale through a scenario-based design approach, increasing the prospects of HCI as a design-science. However, there have been other arguments and methods published within HCI literature that can be valuable. We review three additional general approaches: establishment of design spaces and taxonomies, critical parameters and performance targets, and a reference task agenda.

The Task-Artifact Framework. In the late 1980s, Carroll introduced a proposal for a systematic method to reconcile contrasting perspectives of hermeneutics and theory-based design [26]. This method was founded on the conjecture that successful HCI designs embody an assortment of psychological claims, determining the system's usability. In carrying out an analytical investigation for understanding a design in psychological terms, the task-artifact framework helps designers recognize tradeoffs implicit in the design as users form a goal, act toward its achievement, and evaluate progress. Articulating these tradeoffs as useful generalizations for future design work provides a mechanism for generative problem solving and design, integrating theory development with design evaluation [29].

In the description of this process, Carroll notes that this tradeoff evaluation provides a method for *mediated evaluation*, a compromise that allows explicit goal formation in early stages of design, intrinsic evaluation and modification of goals throughout the design cycle, and inclusion of goal analysis in payoff evaluation [26]. In later work, Carroll argues that the task-artifact framework, coupled with the use of scenarios to articulate user concerns and interface usage, provides a basis for an action science in HCI through the deliberate management of tradeoffs made explicit and assessment of basic tasks [28]. Based on the task-artifact framework, Carroll has developed a gradient of progressively powerful analysis techniques, starting with basic scenario-based design and task coverage through Norman's stages of action (as described previously), and extending to the process of claims analysis and hill-climbing, a taxonomy of concept relations for mapping problem and design knowledge, and object-oriented design methods (e.g., class hierarchy generation and object point of view analysis [27]).

Carroll argues that, during any design process, many compromises are often made, but *claims* concisely articulate the positive and negative effects (tradeoffs) of a feature on a user in accomplishing a task. Claims address a variety of situational and interface aspects that affect the compatibility of the design and user's models, such as user satisfaction and feeling of reward, color and object layout, and strength of affordances. To ensure interface usability, designers can focus on developing and validating key claims associated with essential tasks to be supported by the interface. The process of making claims about the problem context, the general activities addressed by the interface, and the information and interaction design techniques is called *claims analysis*, a design method for mediated evaluation [29] that produces a testable and refutable record of design rationale. In this manner, claims list a set of hypotheses about a scenario or design artifact and "open up a process of critique and refinement" [23].

Claims analysis can act as a hillclimbing heuristic. Carroll refers to *hillclimbing* as a process of achieving a progressively better design solution based on knowledge attained from previous efforts–a collection of existing claims forms the slope that has already been traversed and provides a basis for continued advancement [24]. To hillclimb, a designer focuses on mitigating downside effects of key claims through new design iterations while enhancing or maintaining upside effects. The foundation is improved as auditable claims are strengthened with increasingly compelling evidence derived through theory, user testing, and field study observation.

While interesting as a proposed approach to HCI research, the concept has not been operationalized with extensive tool-support or widely tested. Our discussion shifts to other general approaches toward enhancing HCI research organization, but we revisit Carroll's concepts in the next section.

Design spaces and taxonomies. Many design domains within HCI establish a formal view of their "design space" and taxonomies of important concerns. The most effective design spaces seem to be those that are suggest design trends, design deficiencies, or inspire new design approaches. One example is Card, Mackinlay, and Robertson's design space of input devices [22]. Just as Mackinlay developed a method for determining task-related graphical encoding effectiveness and expressiveness [79], they were able to describe the semantics of input devices with a movement vocabulary and composition operators. By charting the various combinations of input device movement according to abstract tupels (e.g. manipulation operators, input domains, device state, resolution, output domain) and domain-relevant connections, they are able to depict a design space that compares input device features. They further demonstrate how human performance theory (Fitts' Law) can be mapped the design space to portray device performance features like pointing speed, precision, error rates, and so on. As they point out, the design space provides a concise representation of user needs that are matched to available products, which exposes research holes and inspires directed innovation [22].

Critical parameters and performance targets. Another compelling argument for increasing research organization and promoting deliberate design is presented by Newman in a series of related articles. His efforts began with a survey of 1989–1993 CHI Conference papers, in which he analyzed these research products to determine how many built on previous work to create better models, solution techniques, or tools and methods. Finding that only 30 percent of CHI articles fell into this category, he also surveyed five other engineering research fields, finding there that over 90 percent of published work built on previous efforts [96] (it is unclear how he would classify his own article). In a later article, he attributes this phenomenon to a lack of performance parameters held in regard by the research community. By this time, he also notes the tendency within research literature to introduce new systems that provide no progress in clarifying concepts, determining design tradeoffs, describing user tasks, or developing predictive modeling techniques. He emphasizes that without the ability to compare research results, communities are unable to accumulate knowledge or judge whether a design is "better" or "just different" [97].

As a solution, he proposes *critical parameters*, or figures of merit that transcend individual projects to apply to a larger domain, allowing several key aspects of research utility. He states that critical parameters, as established, quantitative measures of whether a design met its purpose, have several characteristics:

- Invariant for a particular design problem,
- Allow description of a particular application or iteration in comparative terms,
- Assessed with direct, empirically gathered data that can be influenced and managed through design,
- Describe crucial elements of a design's purpose,
- Generally taken for granted, since they are universally agreed upon and self-evident in definition [97].

To illustrate the idea, he discusses possible critical parameters for traffic intersections. Although traffic intersections can have many different forms (simple crossroads, signaled intersections, cloverleafs, or fly-over junctions), they all support the same basic tasks and can be measured according to critical parameters, such as *flow rate* and *delays*. Various options can be quickly compared, since they are understood according to these universal measures. Presumably, when traffic engineering develop new implementations, they can be tested and categorized according to these parameters [97]. He emphasizes the point that critical parameters should promote innovation and creativity, but in a controlled, responsive manner [98] (like design space holes). Performance targets within the critical parameters provide this mechanism.

A reference task agenda. Finally, Whittaker, Terveen, and Nardi present an excellent argument for a reference task research agenda within HCI [133]. This argument is founded on the premise that the radical and wild innovation (while currently valued within the CHI community) is not a healthy primary research motivation and has not aided the development of the "science of HCI." Instead, they propose an agenda that provides a structured approach to identifying design requirements and sharing research findings through empirical analysis of essential user tasks, systematic analysis of user interaction, and comparison of measured task performance with critical parameters. *Reference tasks* are common user tasks within a research field that can be described by general problem definitions and user requirements, measured by standard experimental tasks, datasets, contextual information, and metrics, but instantiated by an unlimited number of design variations. Most importantly, in the selection of reference tasks, the research community is acknowledging research problems that are deemed worthy of sustained investigation. Their argument summarizes the positive benefits that resulted in the speech recognition and information retrieval research communities from the adoption of reference tasks.

2.4.3 Software and knowledge reuse

The logical next step after achieving research organization seems to be using the structure to leverage design cycle efficiencies, such as software and design knowledge reuse. Although much more momentum in this area currently exists in the software engineering community than within HCI, it is a topic of growing interest and one that should someday be of great interest to notification system designers.

Describing design ideas. An essential prerequisite to reusing a design idea is that existing designs must be represented and stored in a way that others can understand. Many different methods have developed over recent years, to include Entity-Relationship diagrams, Data Flow Diagrams, and others. Techniques have also been developed to especially apply to user behavior representation, such as the User Action Notation [61]. Work within this area has even established a model for discussing and evaluating behavioral representation techniques [31].

Reuse approaches.

The most dominant approach to software and design knowledge reuse seems to be the patterns movement, coupled with Unified Modeling Language (UML) descriptions. Patterns are a common vocabulary for expressing concepts that help to solve recurring problems. Initiated within the architecture field [3], the concept of patterns has taken hold within software engineering and is gaining popularity within HCI design fields[2]. Since patterns can include records for design tradeoffs that are observed through actual use, they support reasoning about design decisions in a similar manner as claims from scenario-based design. Other record fields allow a prototypical example of a pattern and context information to be included, so that designers can understand the conditions under which the artifact was used. As patterns are being directly associated to component code through UML, this concept holds a great deal of potential for the future of interface design.

Other related work within HCI has focused on developing theories and methods for design reuse in the requirements generation stage [118]. Sutcliffe argues that HCI research should focus on producing "designer digestible" packets of HCI knowledge in the form of claims, grounded on good theory and allowing general reuse. As part of this work, the Domain Theory provides a structure of abstracted domains, interaction sequences, and tasks that can be used to catalog design information. Domain Theory provides a roadmap that is extendible to any design domain. As a key part of scenario-based design, the Task-Artifact Framework, and an extension of Domain Theory, claims provide a strong basis for reuse, but they must be "factored" or made generic in order to achieve a broad enough level of abstraction to be frequently accessed. Sutcliffe and Carroll have demonstrated techniques for factoring claims, as well as the resulting potential for cross-domain design reuse [116]. Within his work, Sutcliffe also provides formal definitions and useful metrics with which to evaluate requirements reuse approaches, as well as a wealth of advice for implementing a reuse program.

Having reviewed foundational literature from HCI and the emerging notification systems research field, in the next chapter we continue with additional assessment of previous work directed at understanding information design for notification systems.

[2] see http://www.hillside.net/patterns/ for examples

Chapter 3

BACKGROUND WORK: EVALUATING SECONDARY DISPLAY ATTRIBUTES

This chapter discusses our initial work in evaluating information design for notification systems. A series of experiments is described investigating the ordering of visual primitives for quantitative data displayed in a divided-attention situation. These studies are presented here in a somewhat condensed form, although full versions are available [37, 120]. This portion of the research is *not* primarily presented as an effort to advance understanding of notification information design, rather, it is the first part of a case study on effectively capturing and supporting the reuse of design knowledge. Inclusion of this work should also acquaint a reader that is unfamiliar with empirical design research to some of the basic concerns of notification researchers.

This portion of the work also serves to motivate the next chapter, which introduces a new research framework for notification systems knowledge. While the execution and documentation of our empirical research followed suit with others discussed in the previous chapter (especially [85, 11, 80]), we were often left with several generalizability and extendibility concerns, despite efforts to enhance the realism of the testing environment. Furthermore, in considering actual notification systems, it is difficult to convincingly apply similar lab-based empirical methods in a manner that can add knowledge to a cohesive research effort. These concerns are fully presented in the final section of this chapter. With the hope that this design knowledge can be useful in a design process, we demonstrate how the knowledge obtained through these studies can be captured for later use in the form of a claim (a concept discussed in the previous chapter).

3.1 Effectiveness of Visual Attributes for Secondary Displays

In a series of experiments, we[1] investigated whether established display design guidelines for focal images (in particular, Cleveland's ordering [40] and Mackinlay's recommendations [79] of visual

[1]Experiment 1 was a group effort which also involved David Tessendorf, Jon Pryor, Ali Ndiwalana, and advisors Scott McCrickard and Chris North. Administration of experiments 2 and 3 was assisted by Jacob Somervell.

primitives for quantitative data) can be extended to information displayed as a secondary task in a dual-task situation. While we were uncertain if Cleveland's ordering would still be appropriate, the ordering had not be empirically tested in previous work. Gauging the applicability of existing design principles is paramount as notification system researchers consider the work that must be done. Unfortunately, we found that the previous recommendations do not hold when value is simultaneously placed on minimizing primary task degradation and comprehending secondary information. However, the experiment results provide a new ordering recommendation for secondary task image attributes according to human cognitive ability to extract information.

Eleven visual primitives (and variations) were tested in a series of three experiments: position along a common scale, size, density, angle, length, color luminescence (for red, blue, green, and yellow-orange hues), greyscale, and unencoded numeric values. These encodings were used in a secondary image that appeared while participants played a block catching game. Secondary images provided information necessary to answer either min/max, comparison, or ratio estimation questions. The first experiment tested position, color (red), and area in both a dual-task and secondary image-only context, providing a baseline for the subsequent tests. In the second experiment, the other hues and greyscale were tested with the same red and position conditions using an identical dual-task platform. The third experiment also included the red and position conditions, but added density, angle, length, and the unencoded conditions. A total of 231 participants were used in the three tests (93, 72, and 66 respectively).

The first subsection discusses the motivation for these studies in additional detail, as well as the experimental platform and design that is common to all three parts. Additionally, I present results, analysis, and feedback from this first effort. The second subsection details the second and third experiments. Some additional related work and minor experimental design variations are provided, but the majority of this material is overall results and conclusions. The most important lesson learned from the series of experiments is that guidelines or recommendations for secondary display information design are highly dependent on combined consideration of the amount of primary task degradation that can be accepted, the importance of secondary information comprehension levels, and the information extraction task supported by the secondary image.

3.1.1 Experiment 1: Position, color & area

Introduction

Computer applications designed to allow user information monitoring and awareness potentially lend enormous efficiency gains within many areas of business, education, and daily life. Displays for multiple or dual-task situations are necessary for applications as simple as instant messengers and news delivery agents, or as vital as vehicular displays, laboratory and security monitors, surgery support and military situational awareness systems. However, ineffectively designed display interfaces, especially dual-task displays, can inhibit, rather than enhance, task performance.

Cleveland and McGill provide an accepted guideline for the presentation of visual data in quantitative tasks, founded on psychophysical theory and experimentation [40]. They recognize visual data as elementary perceptual tasks, described as graph attributes, some of which convey information better than others. Attribute effectiveness guidelines facilitate design of display interfaces that effectively communicate information and create insight. Cleveland and McGill provide an ordering of graph attributes: position along a common scale; position along nonaligned scale; length, direction, angle; area; volume, curvature; shading; and color saturation. Mackinlay extends this list to capture nonquantitative data [79], resulting in the inclusion of more attributes and orderings for nominal and ordinal data. These experiments establish an ordering of perceptual tasks for print-based media, providing a solid foundation for evaluation of graph design in the focus. It has not been established that the same ordering applies to computer displays or to perceptual tasks required in a dual task situation.

Therefore, we empirically investigate a sample of these attributes in a dual-task system, working toward such guide-line reestablishment or identification. The initial question is whether a user's ability to perform an information extraction task on a desktop computer with certain attribute encoding is different when the task is a single task in the user's focus compared to a secondary task in a dual-task situation. If there are no performance differences, to include introduction of distraction to the primary task in the dual-task situation, then this is an indication that focal guidelines for these attribute encodings are extensible to such a dual-task condition. However, if differences exist, then new guidelines for attribute use must be thoroughly investigated and understood.

Related Work

Cleveland extends his thinking about attribute effectiveness to relate with specific cognitive task requirements and visual operations. He recognizes three types of pattern perception operations that form all operations of physical information extraction from graphics: symbol detection, assembly (or grouping), and estimation (discrimination, ranking, and ratioing) [39]. Detection tasks are best supported by single curves or line segments; filled circles that may overlap hinder detection tasks. He states that color can be used to establish categories that enhance assembly, since assembly is enhanced by symbols that have strong boundaries (non-overlapping area). Using color can also provide quantitative encoding to increase estimation efficiency. Position in relation to a reference grid and dot plots with ordered categories improves all pattern perception operations, especially estimation. He criticizes circular area encoding in general, since it fails to provide efficient detection of geometric objects that convey information about differences of values.

Wickens et al. introduce discussion of fundamental cognitive processes as well-search and compare-"that may be supported or inhibited by specific graphical renderings" [134]. This conception of search tasks seems to include Cleveland's detection, assembly, and estimation-discrimination operations. Compare tasks roughly equate to estimation-ranking and estimation-ratioing. Wickens and Hollands examine relative attribute effectiveness as a function of human ability to conduct a parallel search among color variation, as opposed to more timely decoding of other attribute encodings that are searched in serial [135]. Lohse takes a similar approach, stating that since color is detected and organized in parallel during pre-attentive visual processing, it is a more efficient encoding than area, since shape is detected serially [77].

Among the research on elements of dual-task display that has emerged in recent years, nothing appears to apply directly to an effectiveness ordering of visual attributes. Mori and Hayashi's work establishes peripheral task causes of primary task interference in multi-window systems [93]. Wickens and Hollands discuss primary task performance degradation in dual-task situations in terms of resource allocation to secondary tasks and adaptation consequences for excessive workload [135]. Rock and Mack also examine divided attention with respect to parallel and serial pre-attentive processing [106]. Others investigate properties of secondary tasks. McCrickard et al. find that effectiveness of different types, sizes and speeds of secondary task text displays relate to different levels of performance expectations-either identification, or higher level comprehension and memorability [85]. Maglio and Campbell concluded that constantly scrolling text should be minimized, since it distracts more than text that discretely appears and disappears [80]. Bartram shows effective uses of motion in displays, particularly with respect to information presence signaling, information search and association, and filtering or linking of spatially distributed objects [11].

Experimental Design

In order to empirically test relative attribute effectiveness, a participant plays a simple, yet demanding game on a desktop computer. Scripted, timed events present experimental conditions and record subject performance throughout the experiment. During the game playing, which occurs on the left portion of the screen, a single image with similar dimensions and brightness as the game appears for eight seconds on the right screen edge. The eight-second display time allows data within attended and ignored locations to be reliably and accurately detected [131].

The game playing continues while the subject scans the image for information that contains the answer to a question asked before the round begins. Each instance of the experiment includes eighteen rounds—nine dual task rounds (gameplaying and image viewing) as well as nine focal (gamefree) rounds. Both treatments require viewing images and answering questions. Participants are 93 undergraduate computer science students, who received class credit.

Six versions of the program implement a Latin square experimental setup testing the independent variables (three attributes, two conditions: single (focal) and dual task). Three base versions differ only in attribute presentation order. Each of these three versions provides two test iterations, one that starts with the dual task and finishes with the focal images and the other that reverses this sequence.

Figure 3.1 shows attribute scales and encoding schemes. Game rounds cycle through three different question types (identification of displayed minimum/maximum values, ratios, or comparison counts) for a single graphically encoded dataset. Under Cleveland's classification, identification of minimum/maximum values is *detection*, comparison counts is *estimation-ranking*, and determining ratios is an *estimation-ratioing* operation [39]. Under the search or compare classification of cognitive tasks [135], minimum/maximum identification is a *search* task, while both comparison counting and ratio determination are *compare* tasks.

Regardless of version and attribute encoding, round questions and answers appear in constant

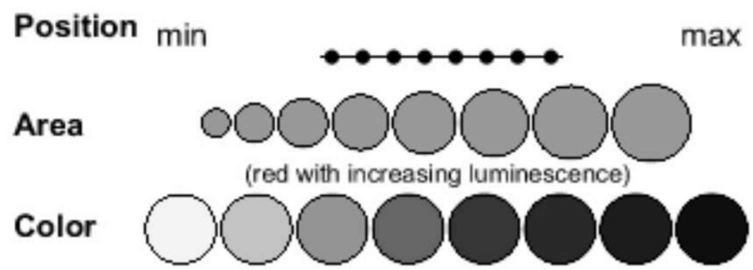


Figure 3.1: Attribute scales and encoding schemes used in the experiment are shown. Relative increases within attribute values are uniform. Subjects are shown the applicable scale before the start of each round.

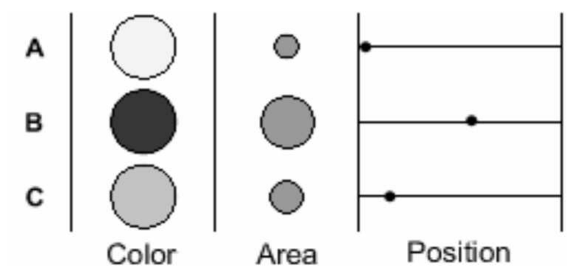


Figure 3.2: Each of the three attribute images (Color, Area, Position) encodes the same dataset values 1, 5, 2. In a given round, one of these images is presented as a secondary task while the participant plays a game. Only three values are shown here, but the experiment's images encode ten.

order. For instance, images like those in Figure 3.2 are used in all six versions as the first graph type, but the question ("what is the minimum value?") and answer (A) do not vary. However, two versions (one version displaying the graph as a focal task, the other as a secondary task) encode this first dataset with position, two others use color, and the final two use area. After testing a single graph with three questions, the dataset and encoding scheme change. Figure 3.3 provides a summary of the experimental flow, as well as screenshots from the actual platform.

Results

This study empirically extends Cleveland and McGill's focal attribute ordering [40] to computer displays-position is best, then area, and finally color ($F(2, 277) = 7.91, MSE = 0.41, p < 0.001$). Specifically, subjects' correctness on answers from the game-free conditions (which only provide the scale and question, the encoded dataset image, and the opportunity to input an answer) correspond to an ordering of position, area, and then color. However, secondary tasks show differences—user ability to gain insight from an image is better when displayed in the focus rather than as a secondary task (answer correctness z-scores range from 13.189 to 1.965, n=93). Figure 3.5 provides a summary of this result.

The next step involves a comparison of answer correctness based on focal images to answer correctness based on secondary task images. In the dual-task condition, answer correctness

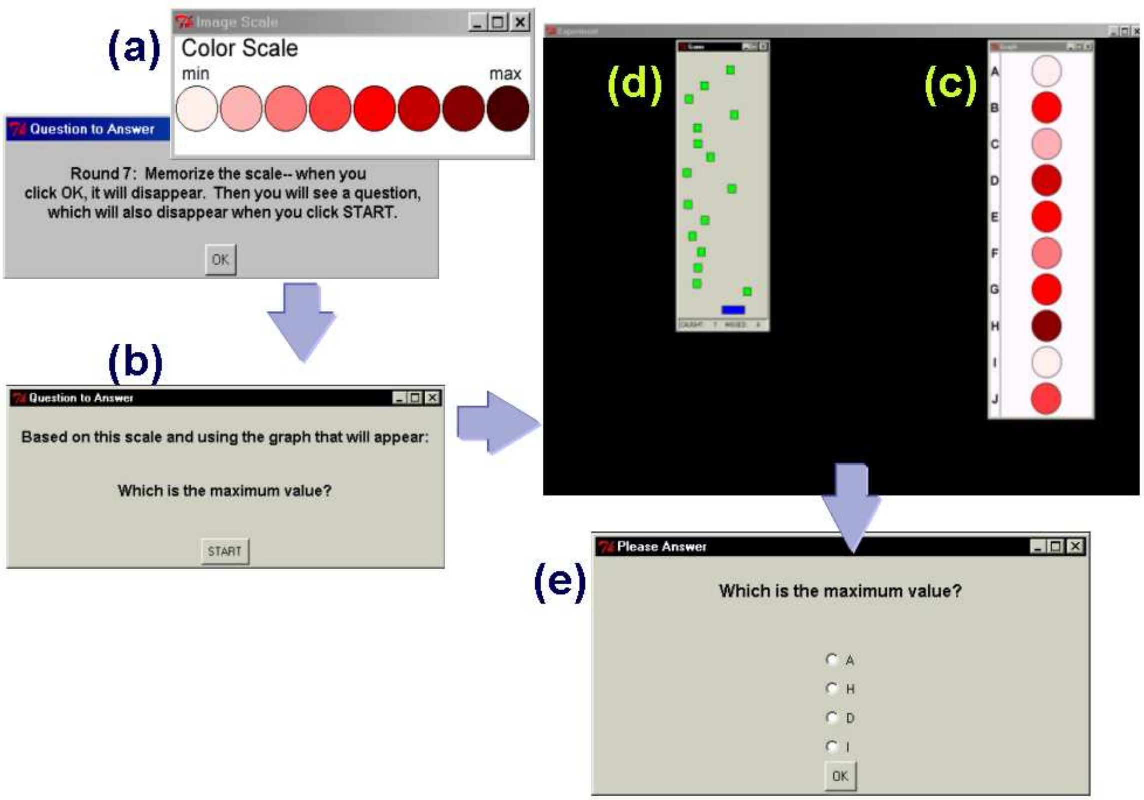


Figure 3.3: At the start of each round, participants are shown the encoding scale (a), and then asked a question (b). Questions require min/max identification, ratio determination, or counting comparison results. Depending on the round type, viewing images (c) may or may not occur while playing a game (d). The game requires a high degree of attention to catch quickly falling blocks. At the end of the round, the question answering dialog (e) captures relative effectiveness of the image encoding and attention-division conditions.

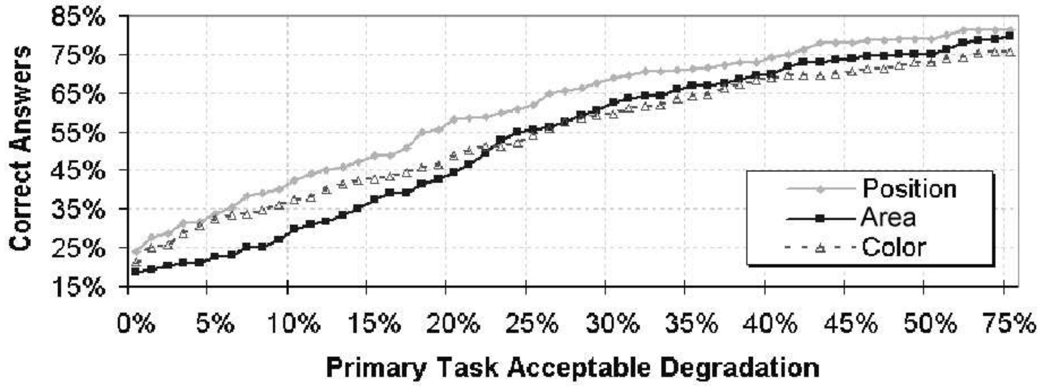


Figure 3.4: Answer correctness by attribute type for levels of acceptable degradation. Note that levels of acceptable degradation cumulate from the left side of the figure (i.e., subjects that meet five percent acceptable degradation include those at zero through four percent, as well as five percent.

requires maintenance of a certain level of primary task (game) performance while the image is displayed. Evaluating a secondary task in a dual-task system with this method, performance effect on both tasks is considered.

> *Primary task degradation* expresses change in game performance during image display period in relation to average game performance before and after the image appears.
>
> *Answer correctness* (secondary task measure) is evaluated for any round meeting a given primary task degradation threshold (*acceptable degradation*).

As a dual-task scoring method, at each level of primary task degradation we filtered qualifying trials (those in which a certain level of game degradation was not exceeded) and examined the percentage of correct answers. Data relating answer correctness to attribute types across degradation levels (see Figure 3.4) is then tested for significance of the main effect, indicating differences due to the three encoding schemes. For all degradation levels between zero and fifty-one percent, there are significant differences in the two results. In other words, given two identically encoded images—one in a user's focus and the other displayed as a secondary task–a user is unable to extract information from the secondary display as effectively and/or without distracting their ability to adequately maintain primary task performance. Similarly, the two activities (extracting focal image information and extracting secondary task image information) are different. The implication of this result is that there is no reason to expect focal guidelines to hold for secondary tasks within a dual-task set-up, when a high amount of attention is devoted to the primary task.

Higher percentages of correct answers always result from position-encoded images. However, at low degradation levels, color-encoded images convey insight more often than area-encoding. The opposite condition is true at higher levels of degradation. Regressed trendlines (sixth order) show significance of main effects in communication of secondary task information encoded with

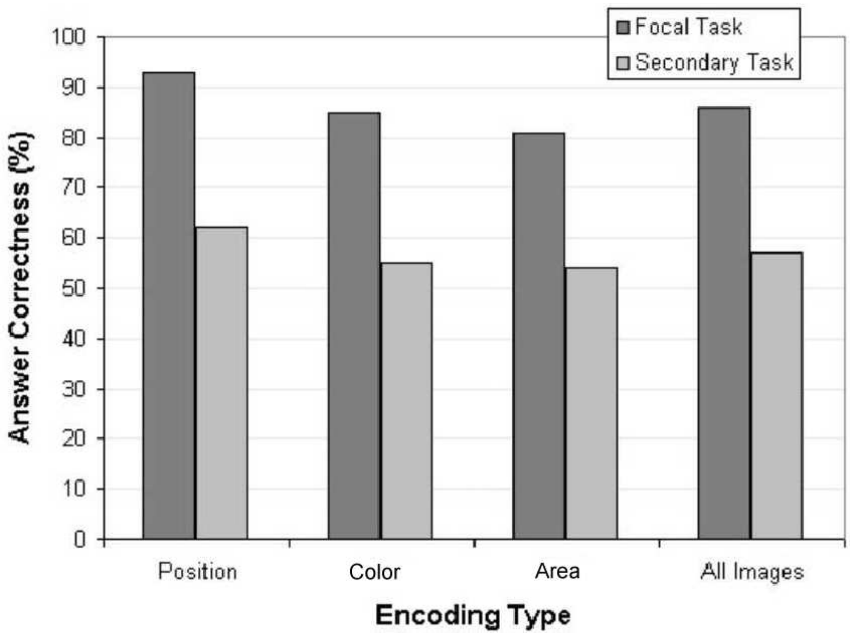


Figure 3.5: Percentage of correct answers for each attribute type, at increasing levels of primary task acceptable degradation.

these three attributes (Figure 3.6). Ordering of attribute effectiveness varies with acceptable degradation and can be completely ordered at low degradation levels—position, color, and then area. Confidence levels for this ordering are established with analysis of variance (ANOVA) tests.

This experiment included three different question types to test information communication from the images. Each attribute is tested with each question, under both focal and secondary task conditions. Therefore, we can find correctness averages according to attribute type and question type, while filtering dual-task performance based on acceptable degradation. This results in a few interesting notes about the various levels of performance for each question type: most test subjects performed best on min/max questions, then count questions, and then ratio questions. Also, count and ratio questions appear to create higher degradation levels in the primary task than min/max questions create. This is consistent with the assertion that Cleveland's classification-detection tasks should be faster and more accurate than estimation tasks. Figure 3.7 depicts these data.

Table 3.1 summarizes zones where significant attribute orderings occur for each question type/cognitive task. "Low Degradation" includes levels of primary task degradation less than seventeen percent, while "High Degradation" includes degradation levels of seventeen percent or more.

Several conclusions are evident from these results. First, considering the superiority of position in all focal and dual task orderings, information should be conveyed in terms of relative position whenever possible to allow optimal probability for accurate communication and primary task sustainability. Secondly, design guidelines—other than those for focal conditions—must address diminished image attribute effectiveness in secondary tasks. Thirdly, it is critical that secondary

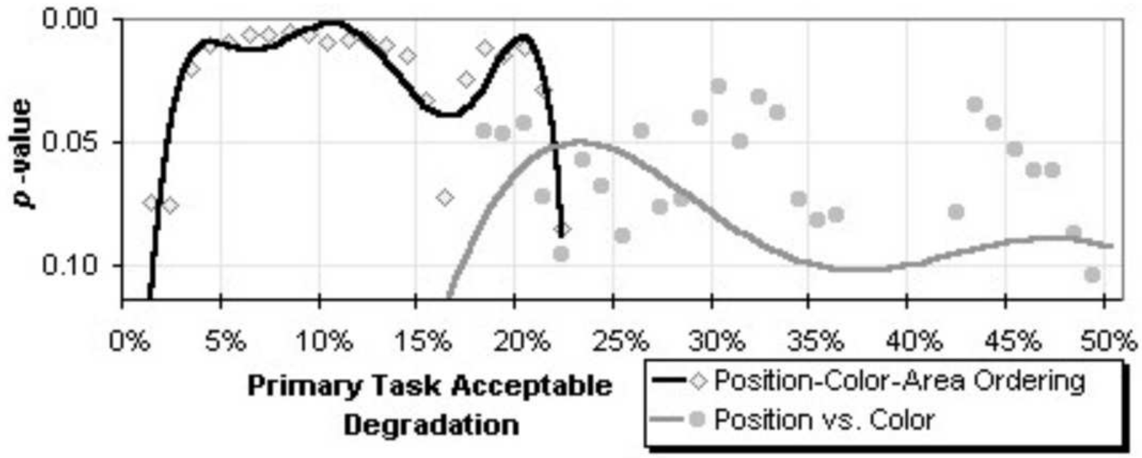


Figure 3.6: Solid trendlines show that levels of significance for attribute ordering vary with degree of acceptable primary task degradation. In rounds with minimal primary task degradation (no more than 23%), subjects answered more questions correctly with position than color images, although both allowed more correct answers than area images.

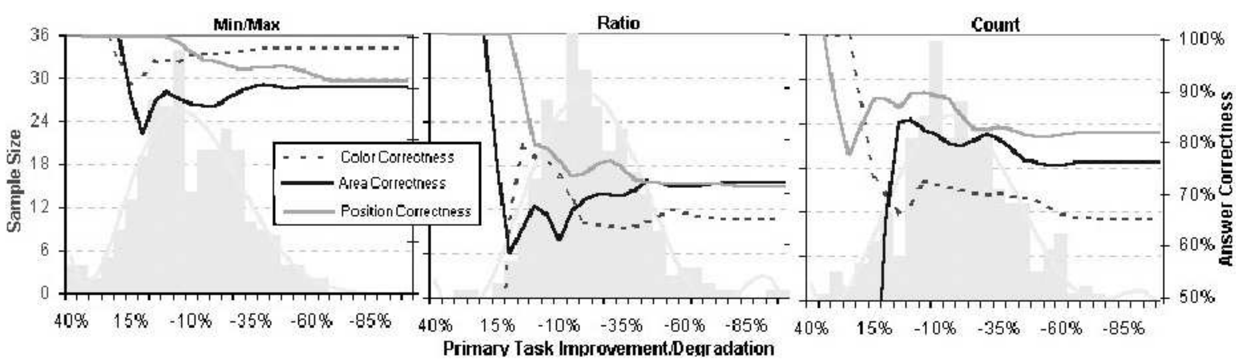


Figure 3.7: Each chart shows information about a single question type. Left y-axis and histogram data show sample size distribution at various levels of primary task performance changes. Right y-axis and trendlines (2-period moving averages) show variation of answer correctness according to primary task performance changes.

Table 3.1: *Significant attribute orderings, by cognitive tasks ($p < .05$). "Low Degradation" refers to trials where primary task degradation was less than seventeen percent, while "High Degradation" was seventeen percent or greater.*

Cognitive Task	Low Degradation	High Degradation
Find Min/Max (*detection*)	position—color— area	none
Determine Ratio (*estimation- ratioing*)	position—color— area	position— area—color
Count Comparison Matches (*estimation-compare*)	position— area —color	position— area —color

task display attributes are selected based on specification of acceptable amounts of primary task performance degradation. Designers of vehicular displays and other systems supporting a critical primary task would certainly want to consider attribute ordering at the lowest levels of acceptable primary task degradation. Therefore, secondary task display design should be guided by relevant attribute ordering in Table 3.1.

Comments & Feedback

Based on the reviewers' comments after submission of this effort to CHI '02, there is interest in an extension of this work to include other display attributes and combinations of attributes in dual task situations. Others were interested in "seeing the results coming out of this research applied and tested in real-world applications." Clearly, the importance of notification systems research was impressed on the reviewer that commented: "dual-tasks occur more and more frequently in daily life, when driving, and, when walking and talking on the mobile phone, such research should be able provide the necessary guidelines in designing the interface of these devices."

This prompted follow-on work to evaluate other visual primitives and combinations of attributes in dual-task situations. In particular, since this experiment limited color encoding to incremental instances of red luminescence, we were curious about the relative effectiveness of other hues, as well as determining whether other attributes may also result in better encoding schemes. This interest motivated the next two experiments.

3.1.2 Experiment 2 & 3: Other hues and visual attributes

Following the first experiment in which position, color (red), and area encoding schemes were tested in both a focus and a dual-task situation, we were eager to compare user performance with other attributes. Since reviewers expressed interest in an ordering of color hues, investigating three other hues (blue, green, and yellow-orange) and gray became our next step (experiment 2). The third experiment tested three other visual primitives (length, angle, and density) along with unencoded text. Both experiments only included dual-task trials, since baseline focal control condition results were established in the first experiment. However, position and red scales were retested in both follow on studies to ensure that these results were replicated, which indicated platform consistency.

The following subsection presents justification and comments related to the selection of attributes and encoding approaches. Results and discussion are presented next, which is followed by brief mention of specific future work that is necessary in this line of experimentation.

Attribute Selection

Although Cleveland and McGill provide their well-known ordering of graph design attributes based on psychophysical theory and experimentation [40], they do not report any specific color results for two reasons: "to avoid the nuisance and expense of color reproduction" and because although Bertin [16] names color hue and texture as elementary perceptual tasks, there is not "an unambiguous single method of ordering from small to large." Instead, Cleveland and McGill suggest that color is best for categorical rather that quantitative data. Of course, Mackinlay refines this ordering to include representation for nonquantitative data [79], listing color hue in the middle of the list for ordinal data and low for quantitative data, although he presents no justification for this ranking. Other classic references for interface design recommend guidelines for the use of color—Tufte, in particular, suggests the avoidance of large areas with strong colors in the periphery, but does not specify this guidance to include hues [123]. Since Lohse, as well as Wickens and Hollands, suggest that graphical encoding schemes should be designed to facilitate quick information extraction by leveraging parallel, pre-attentive visual processing with color primitives [77, 135], a more solid understanding of color hues as they compare to other attributes seems particularly useful for notification systems design. While empirical studies have been conducted comparing information extraction supported by monochrome and color visualizations [62], we could not find specific results by hue, or attribute effectiveness orderings, according to compatibility with secondary display objectives.

To get an initial understanding of performance with different hues, we selected three hues that are very distinct from each other and from red. As was the case in the construction of the original red scale, for each base hue (the fifth shade from the left in Figure 3.8), we used a luminescence value of 120 in MS Paint (ver 5.0), spacing increasing or decreasing steps by 20. The hue and saturation values were held constant for each scale: red at $H = 0$ and $S = 240$, blue at $H = 150$ and $S = 240$, green at $H = 90$ and $S = 120$, and yellow-orange at $H = 30$ and $S = 240$.

In selecting other attributes to include in experiment three, we considered all primitives on Cleveland's and Mackinlay's lists. Direction and slope were eliminated as treatment options due to their similarity to angle. Volume and area are indistinguishable in a 2D representation, so volume was removed from consideration. Three other attributes seemed applicable to graph design, but not to quantitative information encoding in many notification systems: curvature, containment, and connection. Shape and texture seemed to have a wide array of implementation choices that were all nonintuitive—these were eliminated due to generalizability concerns for potential results. The three remaining attributes were all included in the third study. We also decided to test a

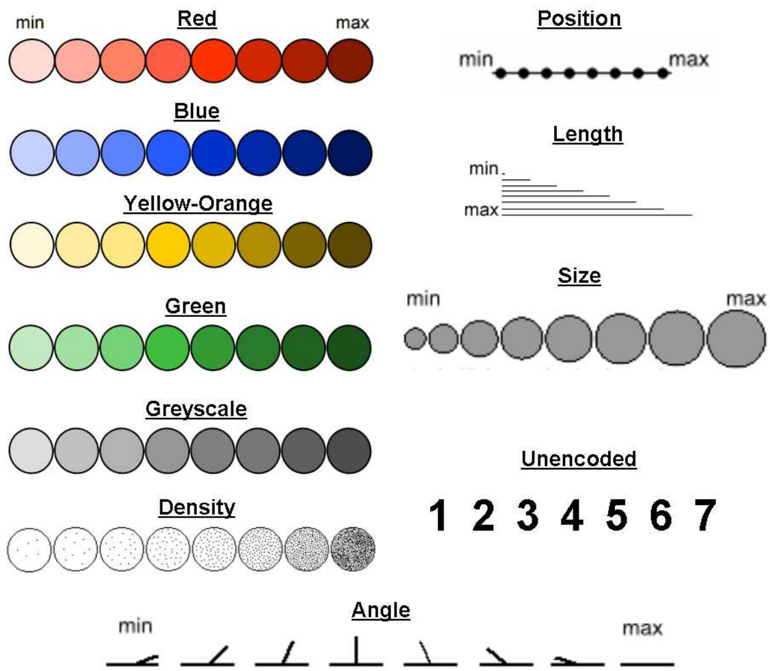


Figure 3.8: The eleven attribute scales used in the three experiments, depicted in scale to each other. Relative changes between values (positions, angles, lengths, luminescence values, etc) are kept uniform.

condition with unencoded (text) numbers as another form of a control condition.

Scales were designed to be as similar to the other conditions as possible, increasing in uniform steps. Figure 3.8 shows each scale. Both experiments used the same testing platform (encoded values for each round, game data, secondary information questions, and Latin square design) as the first one, with only a modification that removed focal rounds. Testing 72 and 66 participants respectively, we distributed participants evenly across each of the six versions. The next subsection presents and discusses results from these trials.

Results & Future Work

Figures 3.9 and 3.10 provide performance rankings for all 11 attributes according to each separate criteria. Certainly, there is no consistent ordering applicable to all situations.

Although these three experiments have addressed the attributes listed by Cleveland and Mackinlay, there are many other primitives that apply to notification systems design that should be included in similar studies. Motion is fundamental—to include grow/shrinking, fading, horizontal and vertical tickering, spinning, and animated color changes. However, these types of motion would not be appropriately tested with this platform, although it is possible that they could be used to represent values of quantitative data. For instance, representing ten values of a dataset would require display of ten motion sequences (varying speed or distance), but simultaneous display of so much animation would undoubtedly confound results. An alternate testing platform should be

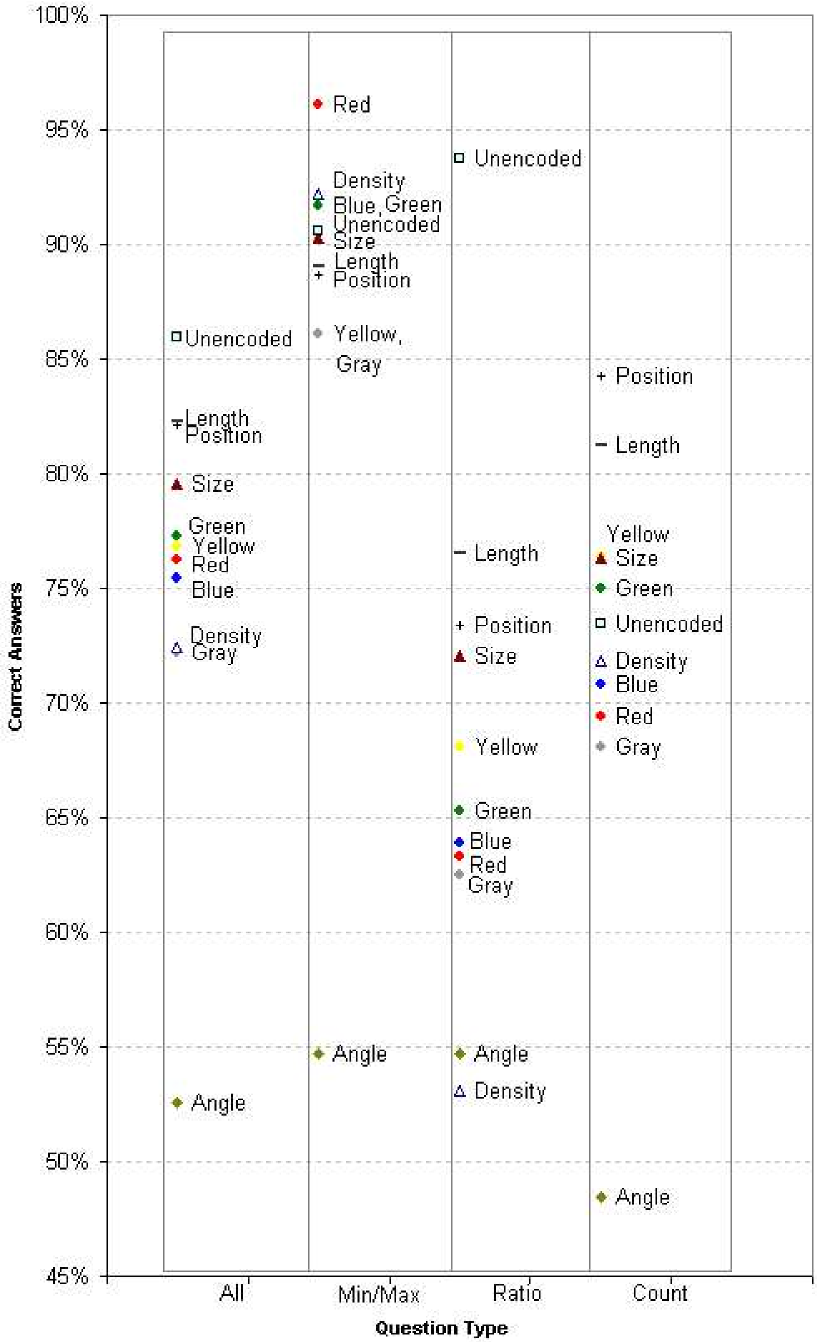


Figure 3.9: Ordering of attributes for supporting communication of secondary information (indicated by answer correctness). Does not consider effect on primary task performance.

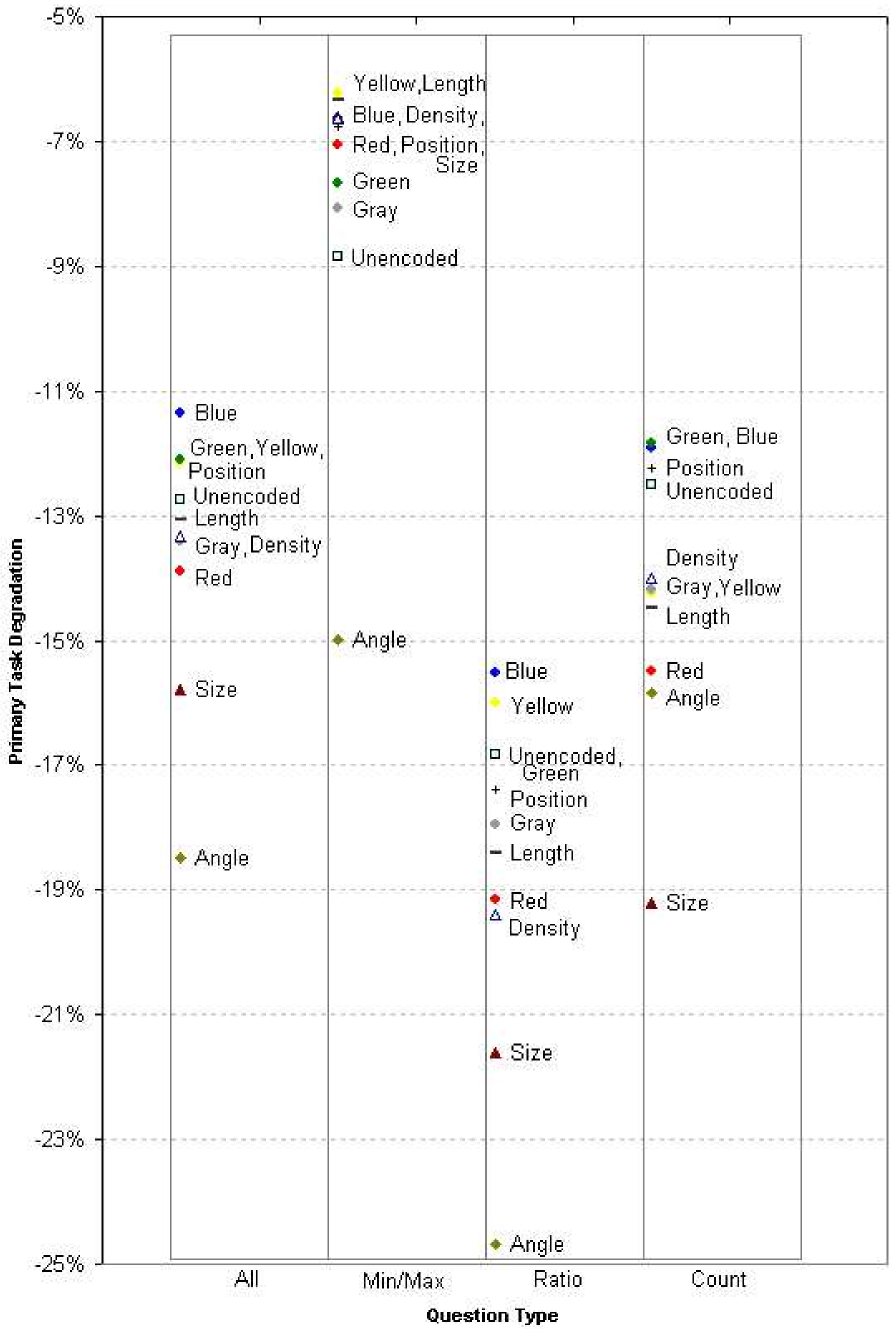


Figure 3.10: Ordering of attributes for sustaining primary task performance (indicated by change in game performance while image was visible). Does not consider support for communication of secondary information.

created to test ability to perceive and decode a single value with various notification formats in a dual-task situation.

With such a platform, various combinations of attributes and implementation options should also be tested. For instance, mixtures such as color-coded text, various sized position indicators, or angles with changing side lengths could allow minimal primary task interference *and* communication of notification information. Likewise, tasks will impact results—just as the attributes tested showed differences in supporting types of information extraction tasks. Our test population was largely young male computer science students, which may be different from average users in many respects. Of course, another consideration for secondary display information design possibly relates to the primary task: information representation, task characteristics, and relationship between the dual-task information are some likely factors. With so many possible attribute combinations, data types, user characteristics, information extraction tasks, and variations between the task relationships, this is quite a fertile area of research.

3.2 Design Knowledge for Design Process?

Reflecting back on the research question and vision for this thesis, we are primarily interested in supporting recognition of whether usability requirements have been met, design usability benchmarking, and design knowledge reuse. As the first step, we apply Carroll and Sutcliffe's idea of archiving design knowledge in the form of a claim (as reviewed in the previous chapter) to summarize the results of our empirical studies. While this initial line of research has yielded interesting information design results for notification systems, as we consider how to show an accumulation of research progress and create other claims, an important problem is exposed-how should research results from multiple experiments be organized for effective reuse?

3.2.1 Archiving study results as a claim

As discussed in the previous chapter, design knowledge can be captured in *claims*, or statements about design artifacts that include the positive and negative psychological consequences resulting from use [29, 108]. As Sutcliffe notes, a key benefit in stating empirical conclusions in the concise claim format is the "digestibility" for designers. In part to demonstrate the practice of converting empirical findings into claims, and in part as the beginning of an example that is revisited later, we present some results from the experiments:

Quantitative comparison with position encoding :
Encoding eight different quantitative values using uniform position changes on a horizontal scale...

+ facilitates interpretation of min/max identifications, counting of specific value instances, and ratio estimations degrades by about one-third from primary task interpretation performance (outperforming most other encoding schemes)

+ effectively preserves primary task performance for primary task with high automaticity, averaging less than 12% degradation per usage trial (resulting in about 45% interpretation success)

+ supports above average or higher rates of primary task sustainment *and* secondary information interpretation, compared to other encoding schemes

− BUT, requires about 27% primary task degradation for two-thirds rate of general interpretation correctness

− BUT, requires reference against a horizontal scale

Quantitative comparison with color encoding :
Encoding eight different quantitative values using circular, red symbols that uniformly differ in luminescence...

+ yields greatest dual-task interpretation success for min/max identification under moderate primary task degradation (outperforming position and area encodings)

+ contributes to greater dual-task interpretation success than area-based encoding for individuals/situations that require less attention (less than 17%) to interpret secondary information

− BUT, degrades success with counting specific value instances and ratio estimations to 10% lower than that supported by position encoding

− BUT, requires about 35% primary task degradation for two-thirds rate of general interpretation correctness

With these artifact descriptions and upside/downside effects, we would also include a visual depiction of the encoding schemes, such as Figure 3.1 and 3.2, as well as details about the testing environment, such as Figure 3.3. These claims will be revisited in later chapters, as we explore how they would be enhanced as reusable design knowledge components.

3.2.2 Limitations of empirical design research

As noted earlier, each of the experiments only begins to probe at the design possibilities. Variation in notification tasks were shown to cause differences in information design effectiveness, and certainly other situational factors in the dual-task system (e.g., variations in the task relationships, differences in the primary task characteristics, user demographic differences, etc.) would produce guideline modifications as well. This observation was especially apparent after the second and third attribute studies. Of course, actual system design usually combines several attributes, such as color, text, and positional information, rather than isolated encoding variables. Based on the performance sensitivity observed in our studies, we can presume that the design variations would also have significant performance effects. These facts lead to the realization that empirical results cannot generalize or apply to many realistic usage situations.

If we maintain desire for lab-based empirical studies, these considerations beg for greater numbers of independent variables in each test, in turn requiring greater numbers of participants and longer test times. However, as we moved toward higher fidelity testing platforms (in an attempt to improve generalizability), we noticed even more difficulty in preparing an automated data collection system and gaining access to testing resources that would enable parallel testing of multiple participants. A tradeoff is apparent between testing numerous, important design subtleties and maintaining a practical and valid protocol.

Certainly, this tradeoff makes empirical study of notification information design more daunting, since it would take many years of effort to accumulate empirical data that would support even a small set of design guidelines (which are probably several years obsolete at that point). This has been the practice in other scientific communities that have been able to achieve progress with extendible testing methodologies and reporting of results. However, as we have seen, the notification systems research community is not yet in a position where systems or design attributes are routinely tested—quite far from a practice where testing results can be replicated and extended by colleagues. Although we are able to make this observation, it does not deter our pursuit for empirically informed guidelines for notification design. Instead, we refocus our energies on establishing a research framework that can accommodate comparison and extension of research results—the topic of the next chapter that serves as preliminary research to the proposed work.

Chapter 4

PRELIMINARY WORK: THE IRC FRAMEWORK

The previous chapter presented my early work in evaluating the design of notification systems. While each study provided some insight about information design tradeoffs in terms of impact to the primary task and communication of secondary information, research that was modeled after other work being done in the field, we were left with concerns about the potential impact. It seems that the current barrier to discussing problems and reporting findings in a common, coherent manner among other researchers prevents applicability of proven evaluation methods, generalizability of findings to new designs, and extendibility of previous results in new studies toward guidelines and theories.

This chapter introduces a potential solution to these problems with some initial ideas for describing notification systems (Section 4.1), a framework based on critical parameters providing definition of a design space, an approach for classifying existing and emerging systems (Section 4.2), and action models portraying cognitive trajectories for various usage scenarios (Section 4.2.3). A case study which discusses observed benefits from these concepts is provided (Section 4.3), and other anticipated benefits (Section 4.4) and the approach for a complete research contribution made in this dissertation (Section 4.5) are also discussed.

The majority of the material presented in this chapter has been published.[1] Much of Section 4.1 appeared in an *International Journal of Human-Computer Studies* article [84], although Table 4.1 has been added to this version. The attention-utility theme explained in Section 4.1.2 was included in [84], as well as in articles that appeared in the *Communications of the ACM* [87] and *ACM Transactions on Computer-Human Interaction* [89]. The explanation of the attention-utility theme and associated table (Table 4.2) is from [87]. Sections 4.2 through 4.3.2 are from [89].

[1] While originally published with co-authors, the material included here primarily resulted from my own efforts.

4.1 Toward Unifying Descriptions of Notification Systems

This section presents some initial ideas for *describing* notification systems. Surpassing a simple definition, the focus is on a taxonomy of usability concerns, a unifying design theme, and the identification of key usage tradeoffs. The taxonomy summarizes the range of implementation characteristics, dual-task situation variables, and primary–secondary task relationships. The design theme captures an essential challenge common to all notification systems, motivating the selection of the critical parameters *interruption*, *reaction*, and *comprehension* in the next section.

4.1.1 A taxonomy of usability concerns

Notification systems allow a user to notice the current state of some information as it potentially changes at some expected frequency. These systems can be implemented on various platforms: desktop computer applications, real world interfaces, large screen displays, and certainly with ubiquitous and wearable computers. All of these notification systems can be passive information channels, or *displays*, requiring occasional attention but no interaction. Others may include additional interactive features making them active devices that we call *interfaces*, which use cues to direct user access of additional information on demand. Although much of the research community is concerned with displayed notification systems, people are well accustomed to audio notification devices such as beepers, email alerts, clock chimes, and the like. Similarly, some real world interfaces leverage haptic sensory perception with temperature fluctuations and vibrating notifications in diverse physical forms. Although notification systems research within the field of human-computer interaction often has been primarily concerned with desktop applications that present notifications, we feel that important lessons can be learned from all notification delivery mechanisms.

Whether a notification system is strictly a display or an interface, there is normally a period of time during its use that users do not keep it within their constant attention. For some interface systems and highly-critical displays, this time period may be quite brief–perhaps only a few seconds. However, most notification systems rarely receive attention focus and are used exclusively as secondary displays. These characteristics describe a system's dual-task nature, a notion which is further developed in the subsections that follow. Another consideration common to notification systems is the priority range its notifications usually carry and the corresponding level of user interest. All of these aspects and usage scenarios have potential implications for information design effectiveness making them important to identify so that empirical studies can determine linkages between cases and design tradeoffs. Table 4.1 is included as a summary of the taxonomy composed by the various design parameters and implementation options discussed further in this section.

Dual-task concept

When a user's attention is divided between tasks, forcing concurrent processing or time-sharing, this is a dual-task situation (we presume one of these tasks to be use of a notification system).

Table 4.1: *Taxonomy of notification systems relating design parameters to implementation options or considerations. We suspect that most notifications will be designed and used in a dual-task situation with some expectation of performance loss for each task.*

	Design Parameters	Implementation Options/Considerations
System Characteristics	Physical platform	Desktop computer | Real world interface | Large screen display | Wearable or ubiquitous computer
	Notification mode	Visual | Auditory | Haptic
	Access to additional information	Yes=Interface | No=Display
	Used with divided attention	Yes=Dual-task system | No=Focal system
	Two-Task performance expectation	Performance degradation | Perfect execution
Dual-task Situation	Attention division method	Graded | Discrete
	Overall mental workload	Overload | Filled | Underload
	Data-link dependencies	None | Primary task only | Secondary task only | Both tasks have dependencies
	Target user characteristics	Task skill levels | Perceptual capabilities
Primary-Secondary Task Relationship	Goal relationship between tasks	Unrelated | State of one prompts goal change in other {Primary/secondary is dependant | dependencies shift}
	Goal-priority relationship	Equal | Primary/secondary favored
	Task execution decisions	Unrelated | Information from primary/secondary impacts
	Task automaticities	Both high/low | Primary/secondary automatic
	Task modalities	Cross modality | Intramodal
	Simultaneous resource consumption	None | Perceptual channels | Working memory | Response mechanisms

Wickens and Hollands provide a comprehensive discussion and reference list for dual-task concepts relating to attention, time-sharing and workload [135]; here, we summarize the most important ideas relating to notification systems evaluation, framing additions to our taxonomy and the later discussion of critical parameters.

If both tasks can be performed simultaneously as well as they could be performed independently, then these dual-task concepts are not relevant and the notification system can be evaluated as any other focal display or interface. However, we believe that use of all but the simplest systems will cause some performance decline or degradation in the other task, implying applicability of the attention-utility theme and the evaluation strategies presented in this paper. We also believe that systems of interest to this research field largely are comprised of dual-task systems. The middle rows included in Table 4.1 describe notification system design parameters and implementation options that expand the design space and are discussed in this section.

In a dual-task situation, there is typically one task that receives attention emphasis, which is referred to as the *primary task* while the concurrent task is the *secondary task*. This is not to say that goals corresponding to the primary task are more important or urgent than the secondary task. Wickens discusses two methods of attention division between tasks: *graded* and *discrete* [135]. In graded resource allocation, a portion of attention is consistently devoted to each task, with the primary task receiving a higher portion. Discrete allocation splits a given period of time (presumably on the order order of seconds or minutes) into blocks during which each task receives focused attention; of course, the primary task would receive attention for more time. Nominally, we assume use of the notification system corresponds to the secondary task, for which attention is allocated on a discrete basis.

Other than fundamental differences in possible primary-secondary task relationships, which we discuss next, many other factors may impact dual-task performance and our ability to understand successful aspects of supporting information design. Some of these include total *mental workload* required by the system, presence of *data-linked* task dependencies, and differences within users [135]. There are certainly conceivable cases of dual-task situations, particularly when one or both tasks are complex or urgent, that result in overload–meaning that mental resources are fully consumed and expected performance levels cannot be satisfied or diminish over time. However, other mental workload characteristics may lead to boredom and disinterest, or attentional resources may be completely consumed by the dual-task goals. In dual-task situations, task execution could be constrained at certain points by availability of data, implying that only so much processing or interaction can be performed (representing progress toward a goal) within a given period of time. Lastly, users certainly have differences in skill levels, perceptual capabilities, and context-switching ability–differences which are important in all areas of HCI but may have even larger implications for understanding effectiveness of notification systems.

Primary–secondary task relationships

When the relationships between the primary and secondary tasks are considered, there are a few important factors which may impact design and evaluation of notification systems, including the relationship between the two task goals and the nature of the tasks. These potential design parameters are also included in Table 4.1, since they expand the taxonomy of notification systems. The two tasks can have goals that are unrelated or that are dependent on information presented in the other. For example, users may want to remain notified of weather conditions while they edit a document–two tasks with unrelated goals. However, in a different usage scenario, information in an instant messaging or collaborative status reporting system could prompt a user to transition from a document creation primary task to a spreadsheet modification task. In other cases, secondary information may not cause a shift in primary task goals, but it may influence primary task execution decisions. Ubiquitous navigation systems are a good example of this type of secondary display–based on the route information presented, a vehicle operator may perform a driving task differently.

Each task may also have intrinsic properties that affect the other task and expectations of notification system performance. High levels of user proficiency or repetitive actions may allow one or both tasks to become automatic, requiring less attention devoted to achieve a desired level of performance. Wickens and Hollands describe other task structural factors, such as modality and resource consumption, which may place unusual strain on a user's attention capacity. Specifically, time-sharing efficiency is enhanced when task modalities allow parallel perception (referred to as *cross modality*) rather than less efficient, *intramodal* perception that could result in serial processing or signal confusion (e.g., reading a book while driving is harder than listening to it) [135]. A final concept from Wickens' discussion considers simultaneous consumption of processing resources (perception, working memory, response) by each task. When both tasks require use of the same human cognitive resources, aspect of dual-task performance may decline although no shift in attention occurred.

Table 4.1 provides an overview of the complete taxonomy introduced in this section. Each facet of the taxonomy may have implications for effective information design for the notification system. Perhaps empirical studies can establish commonalities between various implementation options. However, the primary reason that this taxonomy has been introduced and developed here is to illustrate the vast design space for notification systems and the necessity to carefully consider, implement, and report details relating to system objectives and evaluations.

4.1.2 Attention-utility theme

Chapter 2 described several examples of notification systems, and other general systems have been mentioned in this section. While most of the systems are not described by their contributors as "notification systems," they all share a few general goals and can be considered together within a larger, common design space. In [84], we defined *notification systems* as interfaces that are typically used in a divided-attention, multitasking situation, attempting to deliver current, valued information through a variety of platforms and modes in an efficient and effective manner. We also presented an important distinction between notification systems and traditional HCI research, which we call the *attention-utility* theme [87], asserting that it is useful to think of attention as a constrained resource that can be traded for some utility. This utility is enabled by perceiving

Table 4.2: *Attention benefits and costs. Notification system users expect to gain benefits associated with fulfillment of user goals (left side) by sacrificing attention from other tasks. Costs can be exacerbated by factors of the current situation (right side).*

utility benefits

user goal	general goals
identify state changes understand patterns and trends assimilate complex information monitor resources over time gain awareness of collaborators	**Comprehension** *information is related to existing knowledge and stored for future use*
make decisions modify primary task approach provide response acknowledge status	**Reaction** *immediate response to a notification stimulus, with or without shifting attention*
pace daily activities prompt task transition receive urgent/timely information synchronize with colleagues	**Interruption** *intentional and inherently useful reallocation of attention from other tasks*
reduce stress emote humor cultivate enjoyment augment meaning or presence increase feeling of security	**Satisfaction** *overall enhancement and approval of the general computing experience*

attention costs

situation parameter	cost factors
Context	goal relationships of tasks task perceptual-motor qualities data-link dependencies relative tasks priorities interruptability focus/peripheral location platforms and environment
User characteristics	skill and automaticity cognitive and perceptual abilities current overall mental workload sender/receiver roles demographics
Information characteristics	granularity discrete/continuous modality (visual or auditory) complexity representation richness anticipated value synchronization content relevance

additional, valued information while performing other tasks. This attention-utility tradeoff can be stated as follows:

> *The success of a notification system hinges on accurately supporting attention allocation between tasks, while simultaneously enabling utility through access to additional information.*

The attention-utility theme concisely captures the source of scarcity (the attention of the user) along with the user's purpose in using the notification system (utility associated with access to an additional source of information). Certainly this relationship is not smooth and differentiable, but still generally describes the cost of achieving user goals-a cost which reliably yields benefits when the state of a user's attention can be modeled and matched with appropriate information rendering.

Table 4.2 itemizes component cost-benefit factors of the attention-utility tradeoff. Users ultimately use a notification system to gain benefits, which come from specific types of utility. We recognize four general sources of utility, which can result from associated user goals (left side of table). The general goals of comprehension, reaction, and interruption can be thought of as critical parameters-key measures of system success that can be benchmarked to reveal design progress.

These goals are unique in that the user is willing to sacrifice a certain amount of primary task attention in order to achieve them. Other important system features and user needs must be typically supported in user interfaces, to include privacy, reliability, and trust. These features can negatively influence the amount of required attention without providing a distinct benefit that independently motivates system use.

The level of cost, determined by the amount of attention removed from ongoing tasks, may be elevated as a result of the factors presented on the right side of Table 4.2. For example, above average attention cost factors may include a user's lack of skill in perceiving unfamiliar or complex notification information. Unfortunately, cost factors may not carry a constant value across different situations or result in expected benefits. Poor designs may result from a user accepting a certain cost in anticipation of a certain utility without actually receiving that utility. Usually, the attention required for a user to perceive and process a notification is diverted from attention focus on a primary task, but cost only results if primary task performance is negatively impacted. Attention supplied during natural breaks in a primary task can minimize cost. The many cost considerations-and strategies to reduce them-amplify the importance of inferring and leveraging the state of a user's attention and semantic value of the notification for interface design.

4.2 A Design Space for Notification Systems

Thus far, we have articulated a theme that expresses general goals and characteristics for notification systems. We have provided some insight into the challenge and need for better evaluations of notification systems, demonstrated by the slow convergence of usable and extendible studies. This section introduces a new approach for designing notification systems, based on a design space model of user expectations and notification effects.

First, we look at a method of simultaneously describing a design model with multiple critical parameters. This allows us to consider and label general combinations, forming a descriptive and prescriptive design space. Using a simple model of human information processing allows deeper understanding of the regions within the design space through identification of action models. We demonstrate the utility of this novel approach for notification systems classification in the next section by integrating several examples of existing applications within the framework and illustrating how reusable design guidelines are then possible though a claims-centered approach to usability evaluations.

4.2.1 Critical model parameters

In order to conduct meaningful usability evaluations that will allow systems to become progressively better, Newman argues that we first must define or adopt *critical parameters*, or figures of merit that transcend specific applications and focus on the broader purpose of the technology [97]. He implies that well selected critical parameters can function as benchmarks-"providing a direct and manageable measure of the design's ability to serve its purpose"–and indicate the units of measure for analytic methods that predict the success of an early design. Newman provides examples and makes several recommendations for identifying critical parameters that support core user tasks and goals (as reviewed in Chapter 2).

Evaluating and Selecting Options

Our first step in selecting critical parameters for a model of notification systems was to identify key user tasks and usage constraints. We developed a long list for both. Users' notification goals include typical tasks such as receiving information that is more important than current activities (perhaps prompting task transition), regularly monitoring a secondary information source over an extended period of time, becoming informed about timely instructions or information states to advise critical primary task actions. Constraints to notification system use include information complexity and granularity, situational context, available cognitive resources, associated familiarity and enjoyment, and delivery mode and method (continuity and encoding). To reduce the complete collection of tasks and constraints to a manageable set, we employed two processes: 1) separating design model and user's model attributes, and 2) identifying dependencies to focus on root causes. Each process is described in turn.

First, we considered the distinction between two types of information about users that designers should have available in a design process. Using Norman's terminology [100] (as described in Chapter 2), the *design model* describes the designer's conceptual model of the user's background, goals and tasks, and processing limitations. Likewise, the *user model* refers to the conceptual model that the user forms according to expectations and experiences with the actual system. Our thought is that modeling each according to similar criteria would be ideal (allowing easier comparison), forcing consideration to be on *anticipated and actual effects* of an interface artifact on a user–which ought to correspond to user goals. This implies that implementation details (e.g. information or notification delivery characteristics that may impact sense of privacy, aesthetics, and subjective satisfaction) should not be a first-order variable within the model, but should be thought about as a system characteristic, modifiable at some level to accomodate less flexible design requirements.

Second, we looked at the dependencies in our list to determine primary factors and generalize the tasks as much as possible. The biggest challenge in doing this is identifying critical parameters that are "measureable" and "manageable." Much guidance comes from Whittaker, Terveen and Nardi's argument for reference tasks [133]. Since the purpose of our model is to aid comparison of designs that are created to support simliar user goals and facilitate recognition of design progress, we do not want to select critical parameters that cannot be modified by an interface design. While situational context is certainly an important facet in the success of a notification goal, and tempting to include as a critical parameter, designers are often unable to anticipate or address context variables. Therefore, we reserve aspects of context as an essential element of artifact descriptions and claims, but do not include it as a primary critical parameter for our model. Likewise, while user satisfaction and enjoyment with a notification system may be an independent goal, we believe that satisfaction is usually derived from efficient and effective delivery of the notification according to a positive balance of the attention-utility theme [87]. Our current determination not to include satisfaction as a critical parameter may be reassessed with further research.

However, as we inspected the general tasks that contribute to notification utility though "access to additional information," we recognize that user interruption, near-term reaction, and long-term comprehension are the immediate results of such access. More importantly, these cognitive processes are manageable through design choices and measurable in user testing. Certainly, each has received much attention in the multitasking and notification research communities (as we proceed to describe). Each of the three can also be thought about as a guiding force of a design model and the desired or undesired consequence of the information presentation of the user's model. Therefore, they are the root causes of a design's success–the main factors that ultimately cause a shift in the balance of the attention-utility theme. Based on this argument, we recognize three critical parameters for modeling of notification system user goals and system designs: user interruption, reaction, and comprehension.

Proposed critical parameters

Interruption. User goals and usage scenarios for notification systems often have some requirement regarding the interruption of primary tasks. In the context of notification systems study, we define *interruption* as an event prompting transition and reallocation of attention focus from a task to the notification. Some situations, such as driving a car equipped with an in-vehicle information system (IVIS), require that a notification system not intrusively disrupt user attention devoted to a main task. Guidelines established in the area of IVISs suggest defining limited numbers and types of interactions with the displays, restricting the amount that displays change, and limiting the time that a display is present [8, 54, 122, 111]. However, other situations, such as monitoring a nuclear reactor, explicitly call for notification-prompted task-switching. Horvitz's models and inference procedures present some hope for this design objective, an imperative driven by his belief that human attention is the most valuable commodity in HCI [66, 64]. These models are designed to improve notification utility by considering cost of user interruption and introducing notification presentation appropriately. McFarlane describes a taxonomy and empirical study describing the major dimensions and design tradeoffs related to interruption [91, 92]. The tentative guidelines he established exhibit design goal tradeoffs among the coordination methods, although negotiation-based interruption coordination appears to be best for many cases. Selection of information design for a notification system that is driven by inferred suitability of interruption will likely have impacts on the two other design objectives (reaction and comprehension) and affect overall system utility.

Reaction. The second critical parameter we propose is the rapid and accurate response to the stimuli provided by notification systems, an effect which we refer to as *reaction*. Often, notification systems present cues intended to inform the user of information of interest, often requiring them to differentiate between values. As such, several studies have investigated how to improve reaction to notifications using preattentive processing, which considers how information can be assimilated and understood rapidly by using colors, shapes, and motion [47, 58, 59, 12, 13, 11]. Other work has examined moving and changing text as a method for presenting information in hands-off displays, observing the perceptibility and readability of rapid serial visual presentations (RSVPs) of letters, strings, and words [51, 45]. These types of studies investigated rapid reaction to information, yet they did not consider more in-depth and memorable understanding of it, our third measure of notification systems.

Comprehension. While rapid and accurate reaction to an informational cue is important in many situations, often it is also (or only) vital to use notification systems with the goal of remembering and making sense of the information they convey at a later time. We refer to this as *comprehension*. Again, we consider research relating to textual motion as an initial example for studying relative comprehension of secondary display information. Juola found that comprehension of information was comparable when presented as RSVPs and in multi-line paragraph format [71]. A study led by Granaas found that in scrolled displays, larger jumps (four to ten characters) led to better comprehension than smaller jumps (one to two characters) [52]. Kang and Muter, in comparing a tickering effect to a non-animated RSVP effect, found no difference in comprehension for a reading task [72]. Other efforts have focused on evaluation of various attributes (position, area, and color) in secondary displays for supporting information extraction and comprehension as part of tasks requiring detection, estimation-ratioing or estimation-compare [37]. We found that the three attributes are significantly different in enabling comprehension at various levels of primary task degradation.

Notification systems research should focus on exploring balances between the interruption, reaction, and comprehension design objectives. However, most of these studies seem to focus on one, or perhaps two, of these critical parameters, seeking to identify forms of information representation that provide the best support for accepted design tradeoffs. One of the most difficult and important aspects in designing and evaluating notification systems is to adequately and simultaneously consider multiple critical parameters that gauge different outcomes of a single resource. In order for critical parameters to aid value to research, all three should be acknowledged in an evaluation process and have standard representational methods.

In the case here, various levels of interruption, reaction, and comprehension result from and cause changes in attention allocation. Since notification systems are typically used in a divided-attention situation where they are not the main focus of attention, assessing these critical parameters often requires consideration of both a primary task and the notification task. As if conceptualizing concurrent and perhaps conflicting design model objectives and representing them in a user study is not difficult enough, understanding what the evaluation results indicate about the user's model and using this insight to guide iterative prototype refinement can be quite complicated. The various approaches to these problems taken by different design teams make extending knowledge to new applications difficult as well.

We address this research problem by providing a conceptual design space that facilitates focus on exploring balances between the interruption, reaction, and comprehension critical parameters. This will allow all three parameters to be simultaneously acknowledged, regardless of their impor-

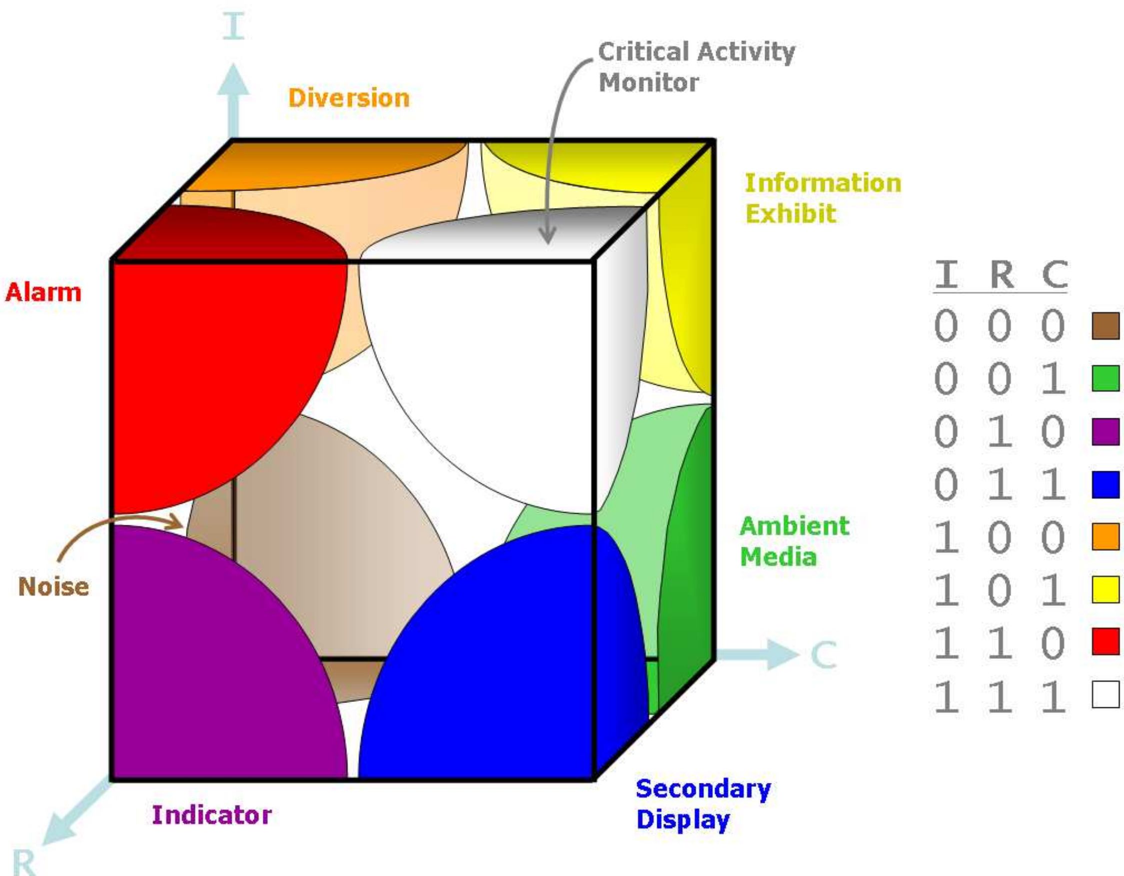


Figure 4.1: The IRC framework, depicting notification systems categorizations according to blend of design model objectives (representing user goals) of interruption (I), reaction (R), and comprehension (C) axes, simplified: low (0) or high (1) and forming a notional cube.

tance within a particular implementation. A successful conceptualization will support visualization of expected usage scenarios and tasks, provide a good fit for all existing systems, and allow communication with standard representational methods.

4.2.2 The IRC classification framework

As a common conceptual framework, we present a characterization approach for notification systems according to their blend of the three critical parameters. The purpose of this is to provide a mechanism that captures the design model–the objective system based on anticipated user goals. To gain a broad understanding, we are initially considering only combinations of *high* (1) or *low* (0) levels of each parameter. For example, a user goal can require a notification system that provides exposure to new information for enhanced awareness over a period of time without introducing interruption to a primary task or prompting immediate reaction to changes. This design model can be described as low interruption, low reaction, and high comprehension, or IRC 001. When

Table 4.3: *Descriptive names, IRC categorizations, and example scenarios for the notification systems design space, forming "scenario families" for notification systems.*

Name	IRC	Example Scenario
Ambient Media	001	An office worker without a window effortlessly maintains awareness of the weather throughout the day with dynamically changing desktop wallpaper. Although knowledge about the weather may be applied in a later conversation or decision, reacting to sudden changes or specific instances is not important.
Indicator	010	A traveler in an unfamiliar city uses a vehicle navigation system to prompt required turns along the route. He has no interest in learning his way around the city, and is only concerned with negotiating traffic and arriving at the destination quickly and safely.
Secondary Display	011	While an editor works on part of a document that is distributed among co-workers, she monitors a groupware tool on the office's large screen display that shows various progress meters for the different parts. Information presented is important for pacing or technique adjustment, as well as an overall understanding of team contributions.
Noise	000	A student working on a slide presentation may not need network access, but perceiving a functional information channel (perhaps providing Internet radio) may be reassuring.
Diversion	100	A home computer user enjoys using his computer more with lower stress if a friendly agent occasionally pops-up with a joke.
Alarm	110	As a businessman attends to various tasks throughout the day, he relies on calendar and email alerts to keep appointments and quickly view important emails. Redirecting activity to the right place, at the right time is the only important consideration.
Information Exhibit	101	A factory supervisor performs routine administrative tasks while maintaining awareness of overall operations. While she expects operational details to be handled by lower level managers, frequent updates are critical for seeing how statuses change over time, allowing assessment of long-term strategy, subordinate decision-making, and operational trends. Understanding this important information often requires close examination due to complexity.
Critical Activity Monitor	111	While performing many routine activities, a system administrator uses a network monitor on a small portion of the desktop. Many users critically depend on his quick and insightful response to network problems, but he is even more valued for understanding specifics or patterns relating to problem prediction and enabling fault-free preventative network maintenance.

we consider this specific parameter combination, it seems to describe an *ambient media*–a passive, "peripheral" projection of information used for meaningfully conveying information status without disruption. Although ambient media has been studied and demonstrated by many other research [42, 60, 69, 81], it has not been effectively integrated into a larger design space.

Extending the same approach to the other seven combinations of parameter levels, it is possible to identify a descriptive name for each design goal, form a useful visualization for regional relationships (see Figure 4.1), and conceptualize user need scenarios (or "scenario families") and identify a descriptive name for each design goal (see Table 4.3). Note that the figure also contains plots of system classifications, which will be discussed shortly. Each critical parameter is represented as an independent, orthogonal axis, increasing from a low to high objective level. While the IRC categorization only precisely describes the corners of this notional cube, we believe it is useful to initially consider regions as extending from objective to near-mid range levels. We fully expect that as this framework is tested and used to describe other user need scenarios, additional logical regions and associated IRC levels will be identified, serving as refined categories of notification systems.

Several ideas may be initially non-intuitive as these design model blends are considered. First, high interruption may appear to be an unlikely user goal for a notification system. However, users often multitask in anticipation or vigilance of the introduction of a certain information state or receipt of a message. For example, in a collaborative document writing activity, a user may be editing a section of the document while waiting for certain actions to be completed by colleagues, maintaining awareness of collective progress with a notification system. When various states of progress are achieved, the user may desire an interruption from the current task that prompts transition to a more important task. Likewise, stock brokers or other decision makers may perform less important activities while they monitor news and stock prices, needing and valuing interruption when important information states are presented–not only to prompt task transitions for immediate reaction but also to enable deeper, immediate inspection of the information. In other cases where interruptions could be valued, users may rely on notification systems to provide advice or guidance for primary task execution–software agents and surgery support systems are compelling examples.

A second potentially perplexing notion implied by this framework is that reaction or comprehension (or both) can occur without interruption to other tasks. At this point, it is important to recall that the IRC levels represent an understanding of *user goals*–the design model–or, objective performance to be facilitated by the system. If designers can leverage skilled memory, task automaticity, and preattentive processing capabilities of users, possibly through use of efficient encoding, rich affordances and metaphors, and cross-modal information conveyance, such design models may be realized.

An important consideration for this conceptualization is the validity in the assumption that these three critical parameters can be considered as orthogonal. Since each IRC blend seems to correspond to potentially realistic usage scenarios and system classifications, this seems like a plausible initial framework. When considering possible usage scenarios for all eight blends within the IRC characterization framework, several differences are readily apparent in design model information interaction approaches. For instance, perception of information changes can be expected to be performed with quick but frequent, non-interruptive glances, careful study during self-defined task breaks, or through peripheral or background perception. Some scenarios called for information presentation that could be fully interpreted and acted on without other information, while others suggest that new information would only be meaningful when associated with previous knowledge or if additional details were accessed.

4.2.3 Notification action models

Norman's theory of action provides the HCI community with a common representation of activity stages required to complete a task [100]. Having this theoretical tool aids the task analysis process, since inspection of interface performance (information or interaction design evaluation) can focus on specific stages or transitions, particularly during a scenario-based design approach [108]. However, when considering a multitasking situation typical of notification systems use with critical parameters like interruption, reaction, and comprehension, the model remains an important influence but seems overly abstract in its ties to cognitive processes. For a theoretical model to be useful for understanding notification systems, it needs to demonstrate parallel processing limitations within and between activity stages, allowing designers to discern conflicts between primary

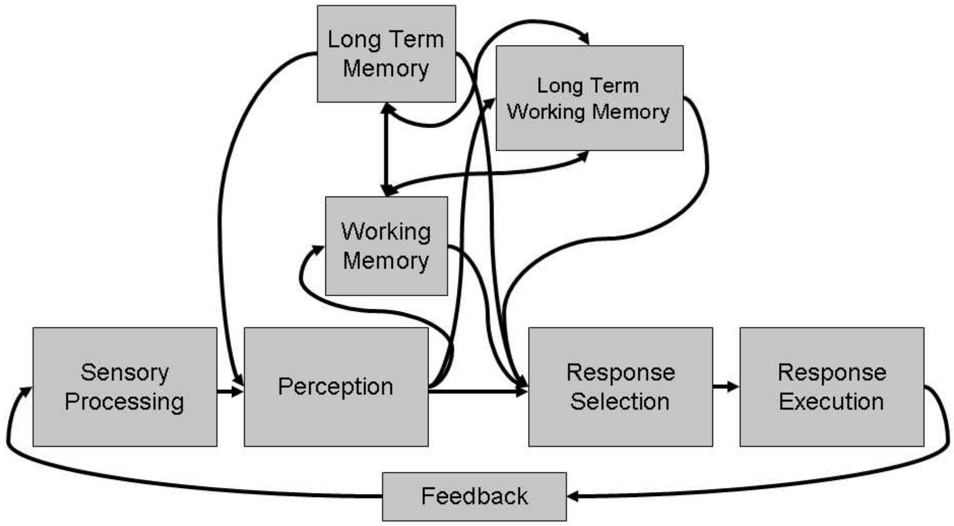


Figure 4.2: A human information processing stage model, from [135].

and secondary activities.

Better representations of task flow should be more closely tied to cognitive architectures, providing both the stage-based focus of a theory of action and the rich link to cognitive science research. Evaluators can target specific areas of the action model for empirical investigation, and seek problem explanations and associated iterative redesign strategies (which Barnard et al. refer to as *microtheories* [10]) from a well established field. Computational cognitive modeling, as demonstrated by SOAR [95] and the more recent EPIC and ACT-R/PM models [74, 4], simulate and predict user performance with interfaces, and may be a long term solution. However, if research, evaluation, and interface design approaches are incompatible with modeling methods, dividends will be slow coming regardless of the model robustness.

Barnard et al. argue that we should consider a system's behavior as "a trajectory governed by systematically structured sets of constraints" [10]. When several systems simultaneously support user goals and resulting interaction, the larger system should be modeled according to a *macrotheory*, with the interaction trajectory providing the center of interest. To form the psychological component of a macrotheory, the authors present a cognitive architecture (Interacting Cognitive Subsystems, or ICS) describing interactors and organization between subsystems that handle sensory input, action coordination, and high-order abstraction of information. This model is quite useful for realizing the processing stages required for and potentially constraining task performance, however it is also quite complicated.

With similar motivation to understand possible interaction trajectories characteristic of notification system design models, we surveyed theories of human information processing stages and found models presented in [135] to be most useful for our purposes (see Figure 4.2). This representation and the related material provided in this reference is particularly handy, since it allows mapping of various trajectories, provides tight integration with our critical parameters, and aids understanding of parallel processing opportunities and bottlenecks.

Using this abstracted model of human information processing, we mapped notification task trajectories for each of the eight broad scenarios for a user's receipt and processing of a notification (discussed in Section 4.2.2). Since arrows depict the possibilities for attention flow, we considered the available flows from each cognitive process that could be used for attention allocation to the notification. As we reasoned about the likely information processing paths for each scenario, we used the associated IRC classification to recall the generalization of user goals. For instance, we can think about interruption as the disruption and resetting of working memory–an inevitable effect of context switch and attending to unfamiliar or complex information for anything but a few seconds. Comprehension requires flow of attention to the long term memory in order to link new information to existing knowledge. Reaction is the observable outcome of the response selection and execution stages. Realizing the presence of each goal (as well as the approximate order that each goal would be fulfilled) ensured inclusion of attention flow to appropriate cognitive processes. The notification task trajectories for each of the eight general scenarios (single notification assumed) are depicted in Figure 4.3.

This provides expected action models for each of the eight design model characterizations. There are several points of interest in this result. First, each path is unique, further supporting our assumption of orthogonal critical parameters. Long term working memory theory plays an important role in our trajectories, and although many ideas are currently debated in psychology channels, the most compelling evidence for this more efficient and less volatile skilled memory comes from dual-task and task switching experimentation [48]. Trajectories for the ambient media and information exhibit categorizations contain a top-down processing element, in which an interface is searched for specific information rather than simply reacting to presentation of stimuli.

If designers are able to gauge user expectations for notification interruption, reaction, and comprehension (forming a design model IRC), they can design the information and interaction display in a manner that promotes ideal flow of attention between cognitive processes, as depicted in the appropriate section of Figure 4.2. For example, it may be argued that in the case of an alarm a user would access long term memory to recall the steps for reaction. However, as depicted with the alarm trajectory, an *ideal* alarm design would attempt to avoid accessing long term memory, perhaps conveying all necessary information in a highly compact manner. As the interface only supports the notification task (and not the ensuing tasks that would be performed after a task switch as primary tasks), this is a realistic design goal. Thinking about the human information processing model in these terms clarifies its usefulness in a design process.

Similarly, when testing a particular design model claim, notification systems evaluators can use these action model trajectories to refer to studies within the cognitive psychology field. Generally accepted testing and reporting methods can be leveraged to capture more precision measures of interruption, reaction, and comprehension. For instance, Rogers and Monsell studied cost of task switching. Not only do they provide an excellent review of related work, but they introduce a method of employing alternating task switch and non-switch trials, effectively arguing that the task switching costs captured describe the need to better switch tasks [107]–essential for understanding usability of systems that provide information guiding primary task performance. Similarly, exper-

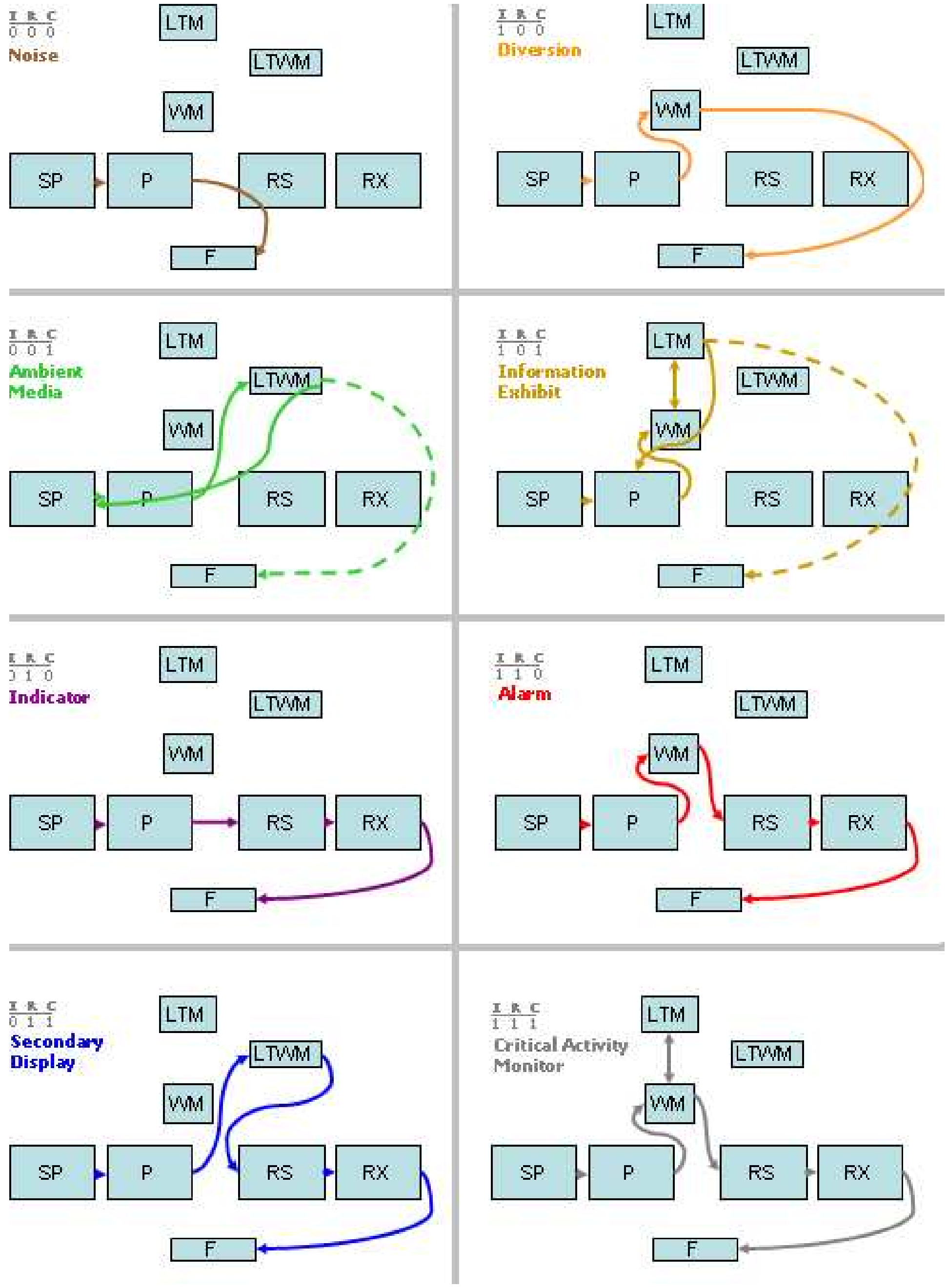


Figure 4.3: Design model flows through the human information processing stage model (see Figure 4.2) for each of the eight main notification systems IRC categorizations (see Figure 4.1), providing a diagnostic tool for thought. Noting unique design path trajectories for each categorization can help identify potential bottlenecks in the attentional flow and progression through the Stages of Action [100].

iments conducted by Baddeley to validate the conception of a homunculus as a model of working memory provide guidelines for dual-task experimentation isolating working memory performance [6]. See et al. provided a review of sensitivity decrement studies for vigilance tasks (particularly useful for evaluating critical activity monitors, secondary displays, and indicators) which not only summarizes important design considerations, but provides a meta-analysis and common view of conclusions from 42 similar studies throughout literature [109]–a feat that seems quite intractable within our field.

Not only can we use these action models to guide our evaluation processes, but understanding concepts such as bottleneck theory (expressed in single-channel theory of the psychological refractory period), cross-modal sensory perception, automaticity, and preattentive processing provide valuable insight for addressing identified user problems. Further discussion of these topics is beyond the scope of this paper, but an excellent review and additional references are available in [135].

4.3 Observed Benefits

In our initial efforts to evaluate the benefits of the IRC framework, we accomplished classification of many different notification systems and design attributes, completed an initial case study of a system's usability testing and redesign evaluation processes, and assessed the potential for the framework as an integral part of undergraduate education. Each of these observed benefits are discussed in turn.

4.3.1 Classifying existing systems

The conceptual framework for notification systems presented in the previous section, formed by the blended critical parameters, can be used as a valuable tool by researchers and practitioners in several ways. Having described a framework for classifying design models of notification systems according to IRC categorizations, revisiting some of the existing systems discussed in Chapter 2 demonstrates how this existing work can be unified. Although it may appear as though most of these systems had very little in common with each other, the attention-utility theme expressed goals common to all of these systems. Each takes different implementation approaches, but they all seek to provide some utility by presenting additional information while appropriately preserving desired attention distribution. Implementation differences are motivated by the designer's expectation of differing interruption, reaction, and comprehension levels desired by a user during their interaction with the notification information.

From the claims made by the authors in describing these example systems, design model details regarding the critical parameters can be inferred (see Figure 4.4). To clarify the method of assigning an IRC classification, we describe our process for four of the systems depicted in the figure:

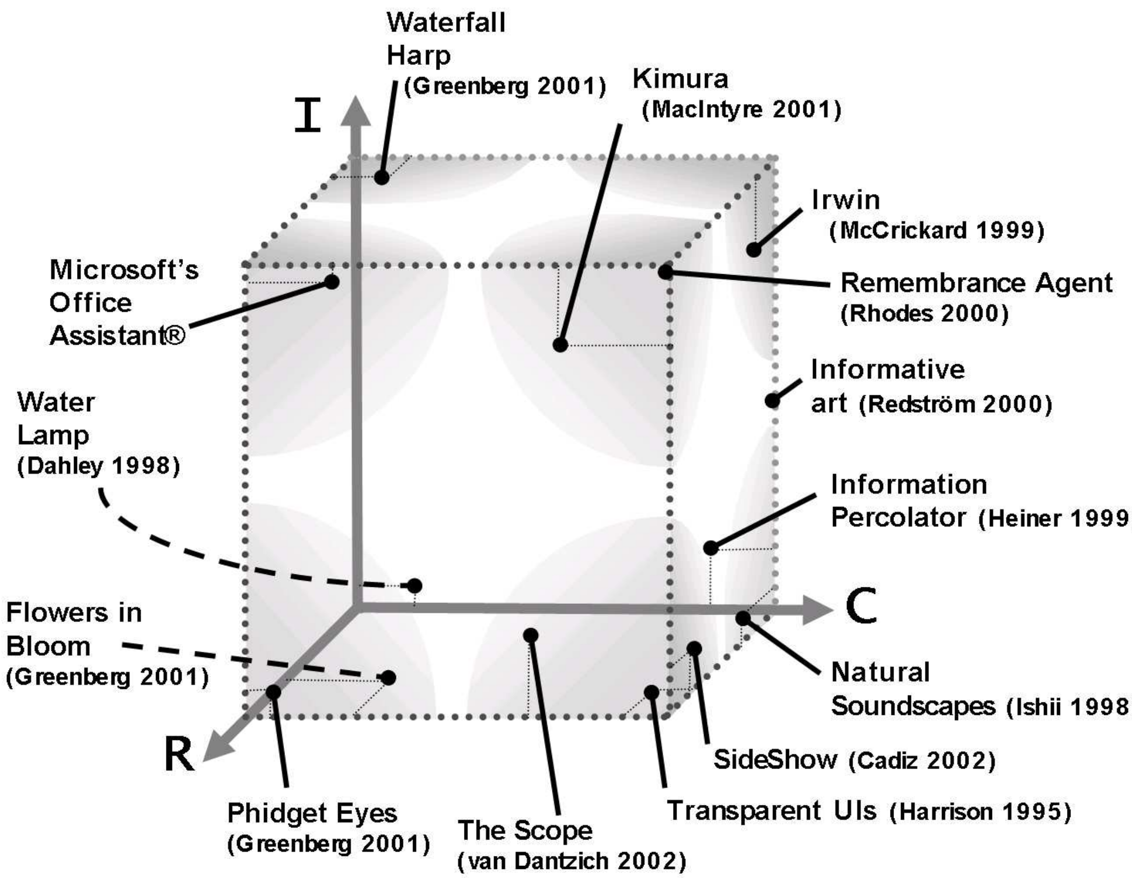


Figure 4.4: Plots of inferred IRC categorizations (design model interruption, reaction, and comprehension objectives) for several notification systems. Plotted locations other than cube corners represent intermediate I, R, or C levels such as "some reaction" or "slight comprehension."

- Informative art–In Redström et al.'s description of these computer amplified, dynamic works of art, they present this class of displays as distinctly different from ambient media or information visualizations, specifically mentioning that these are not intended to reduce information overload by enabling peripheral perception of information (*not low* interruption). Instead, the period of time required to view and decipher deep meaning (*high* comprehension) provides a valued moment of rest and reflection for users (*some* interruption), although the displays are intended to be non-obtrusive, aesthetically pleasing objects during times of non-use (*not high* interruption). Furthermore, no user utility gain is anticipated by prompting responses like spontaneous informal communication (*low* reaction). *IRC characteristic: (.5/0/1)*

- Water Lamp–Dahley et al. provide an example usage scenario for their ambient projection of light through water ripples created by computer-controlled solenoids: enabling a sense of connection to a loved one by displaying their actual heart beat. The projected ripples are intended to be casually perceived and processed at a user's "periphery of attention" (*low* interruption), without invoking moment-to-moment responses (*low* reaction), but providing some awareness of the loved one's activity levels (*slight* comprehension). The true utility gained by a user of this system is anticipated to be added feeling of closeness. *IRC characteristic: (0/0/.25)*

- Remembrance Agent–Rhodes and Maes discuss the goals a user would fulfill with their just-in-time information retrieval agent: as a user types a document he receives an alert (*some* interruption) about related documents with one-line summaries provided at the bottom of the text window. Suggested documents can be old emails, notes, webpages, etc, leveraging and linking existing knowledge (*high* comprehension) or inspiring new ideas for the editing task (*high* reaction). Clicking on the summaries (*high* reaction) allows an easy and desired task transition-access to the full text of the suggested documents (*high* interruption). *IRC characteristic: (1/1/1)*

- Flowers in Bloom–Representing information in a continuum of states according to the bloom-level of an artificial flower arrangement, this device is intended to be non-intrusive within an environment (*low* interruption), providing a single value in each glance (*slight* comprehension) that would facilitate appropriate action (*some* reaction). *IRC characteristic: (0/.75/.25)*

4.3.2 Interface testing and reengineering

To test the utility of the IRC characterization framework, the corresponding action model, and our notion of generic IRC-based UEMs, we conducted a case study. Our case study compared two formative usability evaluations where questionnaires were used as the primary evaluation method. The original was conducted by researchers at Microsoft as part of the iterative design process for their notification system, the Scope. The second was conducted in our lab with a similar study developed using the IRC framework and a simulated version of the prototype. Guidelines derived from the two evaluations were compared to determine which evaluation was more effective.

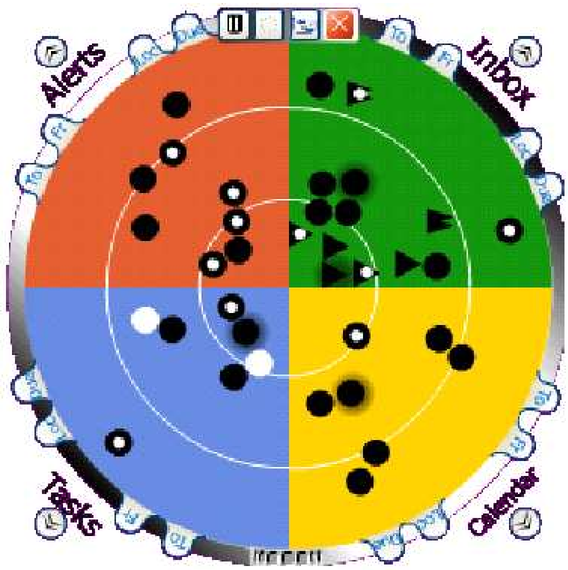


Figure 4.5: *The original prototype design of Scope, the application used as the focus of our case study. Scope sits in a corner of the desktop, presenting notification items as symbols within categorical quadrants. Urgency ratings correspond to centrality within radar metaphor. Scope is fully described in [126].*

System overview. The interface under consideration in both evaluations is the Scope, a notification system developed by Microsoft researchers to help users stay aware of information using a radar-like circular display with higher urgency items located closer to the center of the Scope (see Figure 4.5). The application constantly resides in a corner of the desktop, providing information on and an access point to notifications. The initial prototypes of the Scope divided the space into four categories: the email inbox, a calendar, a task list, and general alerts. The appearance of items in the Scope reflects information such as recipient lists for emails and expired deadlines for calendars and task lists.

According to its designers, the Scope is intended to "direct a user's attention to high urgency items" yet in general require "minimal attention to stay aware of incoming notifications" [126]. According to our IRC model, this means that the Scope should act both like an alarm, supporting high interruption and reaction but low comprehension (IRC 110), and like an ambient display, supporting high comprehension but low reaction and interruption (IRC 001). That is, the Scope is intended to support the alarm-ambient supertask where it must simultaneously enable detection of urgent notifications while facilitating task transition decisions and provide awareness of all pending notifications without distracting other tasks. Scope's IRC depiction in Figure 4.4 as a secondary display represents this supertask characterization.

Two evaluation approaches. After the initial design phase, the Scope developers conducted a pilot usability study intended to identify major usability problems to be addressed in the next design iteration. In the study, six participants performed a series of eleven tasks using the Scope

Questionnaire Item:	**Avg. Rating (1=Disagree, 7=Agree)**
1. The ability to change priorities for an item by drag and drop was easy:	6.33
2. It was easy to move between low and high levels of detail:	6.167
3. The use of pulsing to show "new" items allowed good detectability:	6
4. The ability to expand a wedge was useful:	6
5. I was able to see an overview of my high priority items from the Scope at a glance:	5.67
6. The different shapes of the items were easy to learn and useful:	4.83
7. The Scope Alerts prototype was easy to use:	4.5
8. The different textures/motions of items (pulsing, fuzzy or inverted) were easy to learn and useful:	4.33
9. The ability to delete items from the Scope view was easy:	4.33
10. It was clear what region of the Scope meant "high priority" and what region meant "lower priority":	4.167
Overall Average Subjective Rating:	5.3

Figure 4.6: *Questionnaire and ratings used by the Scope design team, reported in [126].*

in a standalone setting. Tasks included identifying high urgency items that met certain criteria, and interacting with the Scope at appropriate times in appropriate ways. For the tasks, completion times and verbal protocols were collected. After performing the tasks, participants completed a questionnaire consisting of ten questions that participants rated on a 7-point Likert-type scale (see Figure 4.6).

While the general style of the Scope study might be reasonable for traditional pilot studies, it failed to account for the unique interactions users will have with notification systems. In the Microsoft study, participants used the Scope just as they would a word processor, spreadsheet, or visualization tool, and many of the questions on the questionnaire probe standard interface issues despite the fact that the designers claim the Scope is intended to be used quite differently than a typical interface. This seemed to make it difficult for the designers of the study to use the results of the questionnaire in establishing future design iterations.

In our study, participants experienced a similar training base through task completion, but with the added benefits of a dual-task situation to provide a truer sense of the effectiveness of the interface. Rather than using the Scope by itself, our participants kept the notification system running in support of a secondary task, with the primary focus on a document editing task. Participants completed two five-minute rounds, with high-urgency items of interest specified before each round and general awareness questions asked after each round. After answering the questions, participants were informed of the correctness of their responses and reactions to provide them with a sense of their performance. In performing the tasks, participants were instructed that their primary goal should be to complete as much of the editing task as possible while still reacting to certain high-urgency items and staying aware of the general state of the information. We feel that a dual-task situation is necessary to encourage users to consider their behavior given two claim categories: alarm and ambient.

	Questionnaire Item:	Avg. Rating (1=Disagree, 7=Agree)	
Alarm claim	1. The interface effectively supports rapid reaction to important information:	3.17	3.27
	2. It is easy to distinguish important from non-important information:	4.67	
	3. Appropriate reactions to the interface are obvious and intuitive:	3.50	
	4. The interface instilled confidence that all important items were seen:	2.33	
	5. It is obvious when a reaction to information in the interface is more important than continuation of the current task:	2.67	
Ambient claim	6. The interface provides an overall sense of the information:	3.83	4.03
	7. The interface provides an ability to detect and understand clusters in the information:	3.67	
	8. The interface supports easy understanding of how information changes over time:	4.33	
	9. The interface supports easy understanding of links between different types of information:	4.83	
	10. The interface provides these overall understandings of information in a non-intrusive manner:	4.5	
Supertask claim	11. Important new information can be quickly recognized and accessed, and it is possible to stay aware of all information without losing your place in the document:	3.17	
	Overall Average Subjective Rating:	3.61	

Figure 4.7: *Questionnaire designed based on IRC claim categorization (alarm, ambient, or supertask), with ratings obtained in our user study. Apparent from the mean claim ratings (3.27, 4.03, and 3.17 respectively), the Scope facilitated ambient goals best and was most lacking in support for simultaneous (supertask) goals.*

To further enhance the participants' alarm experience, in each round participants were asked to click on specific high-urgency items (such as a new email sent just to you) just as they would when using the Scope in a real setting. In terms of the notification action model discussed in Section 4.2.3, this requires participants to experience stimulus perception, working memory dump, and response selection resulting in task transition. By completing several such alarm-style interactions, participants should be better prepared to judge the Scope's ability to support alarm interaction. To encourage the ambient experience, participants were informed that at the end of the round they would be asked questions about the information that appeared in the interface (such as the total number of items or the category in which the most new items appeared). In terms of the corresponding action model, this requires participants to experience stimulus perception, maintain their working memory, and yet expand their semantic memory with new information. By answering several such ambient-style questions over multiple rounds, participants should be better prepared to judge the Scope's ability to support ambient interaction.

Going into the questionnaire, our participants had experienced a more realistic usage environment and should be better prepared to assess the ability of the Scope to act as a notification system in the ways intended by the designers. Our questionnaire is divided into three parts: an alarm assessment category, an ambient assessment category, and an alarm-ambient supertask category (see Figure 4.7). We developed the questionnaire to be of comparable length to the questionnaire that the Scope evaluation team used. Question selection was based on our assessment of the Scope's design model (as discussed earier in the case study description) and is intended to explore the tradeoffs between interruption, reaction, and comprehension experienced by the participants. For instance, designers of the Scope anticipate users will welcome brief interruption to properly react to sporatic, high urgency notifications. Support for this alarm goal is assessed with the alarm portion of the questionnaire. As we thought about key reaction questions, tenets of signal detection theory outcomes were influential. However, normal use of the Scope is expected to allow longer-term awareness of notification items with glances that do not interrupt the primary task or invoke immediate reaction. This ambient design model is tested with a different series of questions, which probe user satisfaction for support of typical and general ambient notification tasks. Rather than trying to speculate about the combined effect on users that results from simultaneous and disparate design models, we added a final question to test the supertask.

All questions were intentionally designed to be generic so that they could be readily applied to other interfaces supporting similar design models–thus enabling benchmarks and comparison. While continuing work focuses on validation and factor analysis of testing instruments that are adapted for IRC models, our intent with this case study was demonstrate the performance of a testing tool that could be mapped back to the IRC model.

Comparison of evaluation results. To judge the merits of our redesigned evaluation method, we compared the findings from both questionnaires with the actual redesign, which was based not only on the original questionnaire but also on user comments and expert reviews. One concern with the original evaluation was that many of the apparent findings from the questionnaire were not followed in the new design, suggesting that it did not probe the issues properly and it did not provide the participants with a realistic user experience. For example, the third question in the original questionnaire suggested that pulsing of new items for three seconds supports good detectability, which may be true when using the application in a standalone manner but which may not be adequate when simultaneously engaged in another task.

In fact, many of the responses to the revised questionnaire suggest that the alarm functions are not adequate, a feeling clearly shared by the Scope designers who chose to revise the way they highlighted new items but not supported by the original questionnaire results. Numerous other cases of disparity between usability test results and redesign decisions emerged from the original but not the revised questionnaire, such as the decision to keep the radar metaphor, improve the shape distinctions and item encodings, and enhance the glanceability. In addition, the redesign decision made by the Scope team to "improve region color constrast for better item detection" was not motivated by any specific portion of their study. Figure 4.8 provides an overview of all of our conclusions. To summarize, the Scope redesign was based on designer intuition rather than their own usability testing results, while results that were generated by the IRC-based questionnaire provided a close match to the intuition of this seasoned design team.

Our study employs a reusable approach such that other applications can be judged using similar methods. The dual-task usage scenario experienced by participants provides a good model for other studies of notification systems. The questionnaire provides a reusable base that can be applied to other notification systems with design model claims of supporting either alarm, ambient, or alarm-ambient supertask interface functionality for formative and summative evaluation.

In conducting other types of evaluations, the approach we undertook in designing this evaluation can map to other empirical methods or analytic approaches. Our previous work, instrumental

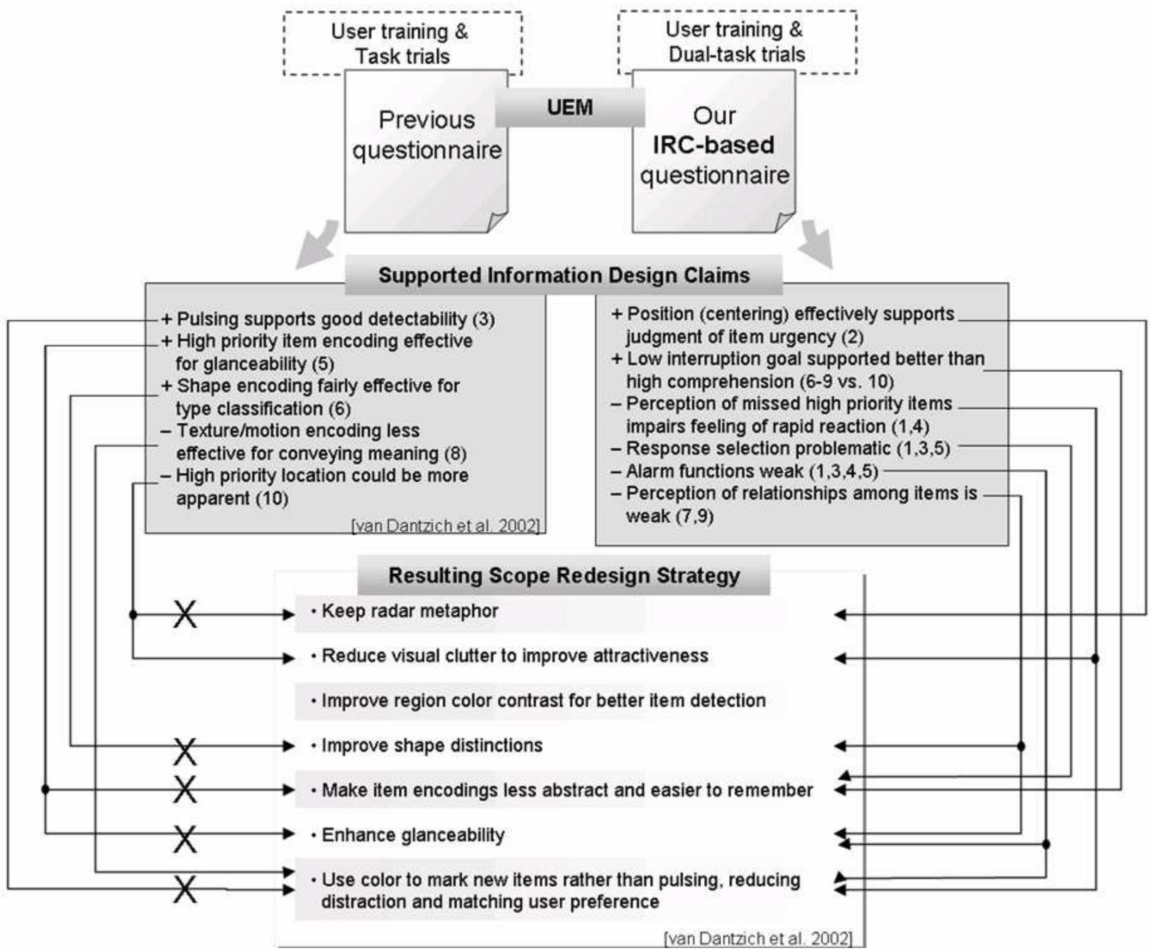


Figure 4.8: *Case study summary. From the previous questionnaire and our IRC-based version, we extracted information design claims and then mapped them (using arrows) to the redesign strategy actually selected and reported in [126]. Note that an (X) on an arrow denotes inconsistency between identified claim and redesign action. Clearly, the IRC-based questionnaire supported the actual redesign strategy decisions better.*

in the development of the IRC framework and notification action models, examined the evaluation of notification systems in empirical studies with primary task degradation, timed rapid response tasks, and answer correctness as dependent variables [85]. In extending to other evaluation styles, it is necessary to provide realistic experiences and probe the use of the notification system according to tradeoffs among interruption, reaction, and comprehension. For example, the primary task degradation used to study interruption in our empirical studies was examined using questions 5 and 10 in the case study questionnaire (see Figure 4.7) and could be explored, say, by observing decrease in productivity during high email periods in an ethnographic study in the workplace.

The advantage of this evaluation approach is that knowledge gained can be directly applied to new design processes, isolating design challenges for iterative refinement while retaining the link to critical parameters. As the area advances, there emerges a cataloging of design models and information design claims, providing a richer base for future notification systems researchers to use for comparison and inspiration.

4.3.3 Teaching undergraduates about notification design

The IRC framework has been used in five semesters (Fall 2002 - Fall 2004) to provide undergraduate and graduate students with a learning framework for notification systems design challenges. We have observed some very positive results, which are thoroughly described in [35]. Certainly, the framework provides a vocabulary and structure that can help novice designers consider and discuss psychological tradeoffs and design options. With a formalism like the IRC framework, we have challenged students to think about appropriate evaluation methods, create portions of a grammar to describe general requirements, and compare effectiveness of design techniques for specific design models. With continued development of the framework and associated tools, our work explores even richer activities that can be developed to introduce students to task modeling, design knowledge hypothesis testing and reuse, and other HCI topics.

4.4 Anticipated Utility

As the general concepts discussed eariler–specifically, the taxonomy of usability concerns, IRC design space framework, and notification action models–are refined, several benefits are expected to emerge for the notification systems research field.

Improving usability evaluation with adapted UEMs. Understanding the challenge allows us to measure the challenge. The articulation of the attention-utility theme and critical parameters provides a much-needed focus to traditional UEMs. Furthermore, the taxonomy of usability concerns can be parameterized to create alterations to UEMs for specific applications. We believe that the notification systems design space, as described by the IRC characterization framework, is sufficiently concise to facilitate the creation of generic usability evaluation method implementations. That is, each region of the IRC framework can have corresponding method implementations that can be used in any evaluation. For analytic methods, this could mean using associated action models to guide a walkthrough process, or using heuristics that are specifically designed to capture targeted levels of interruption, reaction, and comprehension. Regions can also prescribe experimental metrics and procedures, as well as methods for field studies and items for questionnaires that can be used to capture comparable data. We probed this notion with the case study described earlier, and continuing work within our research group will improve our understanding about this anticipated benefit.

Enabling design reuse with claims catalogs. Sutcliffe argues that HCI research should focus on producing "designer digestible" packets of HCI knowledge in the form of claims, grounded on good theory and allowing general reuse. A design space lends organization not only to the classification of systems, but also to components of systems–artifacts and tasks. Since claims can be associated with artifacts and tasks, they can also be organized and referenced according to IRC classifications. New design model concepts can be matched with claims that are correspondingly cataloged within common IRC characterizations, allowing reuse and enhancing opportunities for incremental progress within the field. Assessing the potential and procedures for intra-regional comparisons and claim applications will be more difficult, but will also add immense value to our understanding of notification systems usability.

Promoting empirical comparison and extension. Generic UEMs for notification systems and catalogs of claims implies a solution strategy for convincingly conducting and recording summative evaluations, as well as matching established claims to new applications. Summative evaluations for systems within a common categorization region become simple with generic UEM implementations. Benchmark levels of critical parameters can be determined for reference tasks in due course. This could be quite useful for judging design potential of new artifacts in early development stages, as well as assessing design progress over time.

4.5 The Way Forward: A Complete Research Contribution

Although the IRC framework and its general research approach have been met with a lot of positive interest and encouragement, we have also received critical feedback. Perhaps the most insightful comments come from proposal and conference reviewers–both of which rejected earlier presentations of this material. We were also able to receive some excellent feedback from attendees at a workshop on design and evaluation of notification systems at UbiComp 2002, as well as reviewers of [84, 87, 88]. Some of the issues raised by many anonymous reviewers and workshop attendees have already been addressed and improved in the published versions. However, other comments had not yet been addressed, and were viewed as essential for the continued progress of this research direction. The following list summarizes unresolved criticism:

1. *Any system can be considered to be a notification system–the definition and conceptual models should do a better job excluding systems.*

2. *It could be made more clear that notification systems are a clearly distinguishable genre of systems.*

3. *Both the classification scheme and the method for assigning values is too subjective.*

4. *It's not entirely clear that designers would at any point stop and say 'Hey, we're building a notification system. Let's see what's available to help us.'*

5. *The case for the utility of an accurate classification needs to be stronger.*

6. *Demonstrate how the proposed framework would move substantially beyond the human information processing frameworks, such as those discussed by Wickens.*

7. *The case studies described do not provide a lot of support for the classification system. Overall, despite my concerns about the case studies, I believe the classification system can be quite useful. However, before using the system, I would recommend that practitioners look for more convincing validation of the system.*

While the research presented in the subsequent chapters will not resolve all of these issues, the main intent is to address the most critical ones as soundly as possible. Items 1 and 2 will be addressed in the Chapter 5, improving our ability to accurately and consistently describe a notification system. Item 3 will be addressed in both Chapter 5 and 6, while an improved argument against items 4, 5, and 6, as well as a solution for item 7, should result from the benefits associated with design comparison and reuse, demonstrated in Chapter 7. Since these have been "show-stopping" points in otherwise favorable reviews and positive research momentum, as motivation for future efforts, they seem to be the most feasible driving force for follow-up efforts. We focus on enabling common descriptions of notification systems and their usability, in terms of both the designer and user's conceptual models.

Chapter 5

PROVIDING DESIGN MODEL IRCs TO EXPRESS REQUIREMENTS

Our vision is that the IRC framework, presented in the previous chapter and appearing in various articles [84, 87, 89, 35], will allow an improved usability engineering process and increased research cohesion to emerge for notification systems, lending efficiency to several aspects of a system design cycle. The research presented in the next three chapters provides support for this vision, focusing on three key aspects:

1. **System description,** which would allow widely understood articulation of notification system design objectives that are focused on critical user requirements.

2. **Interface evaluation,** to enable comparison of the design and user's models, while supporting generalizability of research and early identification of usability concerns.

3. **Design comparison and reuse,** that would save time and effort in requirements analysis and early design stages by enabling design reuse and appreciation of design progress–both which are informed by reference task benchmarking.

This chapter describes work done to improve consistency and accuracy of system/requirements description with the IRC framework (item 1, above). Section 5.1 introduces this first project more thoroughly, discussing the underlying motivation, problem statement, and successful endstate, as well as providing an overview of the general approach and expected results. To approach the problem, we started out by reconsidering designers' tendencies to manage IRC parameters in notification design, whether implicitly or explicitly. Findings of this review appear in Section 5.2. Responding to this requirements analysis, our design efforts are detailed in Section 5.3, which culminate with the prototype creation and tuning of a tool, referred to as "IRCspec." Formed largely from material published in the *Proceedings of the Conference on Computer-Aided Design of User Interfaces (CADUI '04)* [32], Section 5.4 discusses the validation (user testing) of design model IRC parameter specification using IRCspec, and the chapter ends with general conclusions about this project, as well as future challenges and directions.

5.1 Issues with Expressing IRC Design Models

Our experience in thinking about dimensions of the notification systems taxonomy and critical usage parameters has impressed us with the complexity of notification systems design. The various facets of the dual-task paradigm may be entirely missed by a casual designer, unless they are knowledgeable about human information processing. Although we have no illusions that designers everywhere will someday use the IRC framework to reason about their designs and gain access to design ideas, we think that the framework can be a useful teaching tool (especially for graduate and undergraduate HCI or other novice designers) and is a step toward creating better design tools for designers and other researchers. If, through use of a deliberate requirements analysis module for notification concerns, students can realize the complexity of the design model formulation process, they may later continue to access other deliberate planning tools to aid early design reasoning before jumping into implementation alternatives. While there are certain to be many ways of classifying a system or articulating design intentions, we believe that developing one possible method will inspire alternatives and prompt discussion of underlying taxonomic issues.

Claim 1 - The IRC framework could allow designers to classify notification artifacts…

+ artifacts are compared by effects on key psychological concerns, summarized by critical parameters

+ full IRC notation carries the *intention* and *effect*

− BUT, no standard process available for deriving design or user's model IRC values

IRCspec, **Chapter 5**

IRCresults, **Chapter 6**

Based on criticism received, we find it essential to help designers realize whether or not the system they have in mind is even a notification system. Since notification systems can be implemented on many different platforms and involve a variety of information design strategies, it will be especially challenging to channel concern to the information need and situational context associated with the notification delivery.

A procedure for considering design model concerns of notification systems may help an individual designer recognize important facets of user background and the usage context. The IRC framework provides a method for reducing a design model to three parameters. *However, the process of obtaining a numeric representation for each parameter value requires consideration of several different factors and is quite subjective by nature.* If design model IRCs are to be compared with each other or interpreted by others (e.g. evaluators, implementers, or other designers), there must be a more standard method of deriving a set of values. Since the design model assessment procedure includes an abstracted consideration of key requirements analysis concerns, it should feed naturally into an IRC value specification. Providing designers with a tool for calculating an

IRC value automatically within the design model assessment procedure could have two implications: 1) the argument about the importance of our parameter selection is strengthened, and 2) any designer can generate a design model IRC without any knowledge of the IRC framework (or even the fact that they generated it!).

Claim 2 - IRC values reduce a notification system design model to three parameters…

+ may help a designer focus on important facets of user background and the usage context

− BUT, its uncertain that notification designers do/should think about IRC effects → Design Rationale Study, **Section 5.2**

− BUT, the process of obtaining a numeric representation requires consideration of several different complex factors and can be subjective → IRCspec Tool Design, **Section 5.3**

In order for the IRCspec system to be thought of as reliable process for expressing a design model, there must be results that indicate sufficient consistency and accuracy in matching a given design conception. Design conceptions can occur in all regions of the IRC framework and include a wide variety of information and platform needs. Therefore, results should include data on a representative sample of notification system design challenges. We especially value data that is collected in controlled settings through conditions that closely simulate actual experience. These objectives form the motivation for this project.

Claim 3 - An IRC design model specification tool…

+ strengthens the argument about the importance of each parameter

+ allows designers to obtain an IRC value without an knowledge of the IRC framework

− BUT, requires proof of sufficient specification consistency and accuracy → IRCspec Validation, **Sections 5.4, 6.4.1**

5.1.1 Problem statement and general approach

Describing user goals with the IRC framework should focus a designer's consideration of user requirements on elements that are crucial to a system's success and allow more accurate communication of the designer's intent. The research question addressed through these efforts is:

Can an IRC rating express a designer's conceptual model of user requirements (design model) with consistent and sufficient accuracy?

To determine this, there were several steps required to augment the IRC framework for facilitating translation of design considerations to an IRC value. First, we needed to consider the results of a more thorough analysis of designers' existing levels of reasoning about IRC requirements in design, specifically asking whether "designers *really* do manage IRC parameters in their design efforts?" The results of this analysis helped us generate specific requirements for a procedure that helps a notification systems designer consider key elements of the usage experience and user's goals and expectations. Second, we developed an automated method of processing designer responses that calculates a design model IRC. Next, we developed a test that in which participants (acting as designers) considered a design problem and completed the design model assessment procedure. Finally, we analyzed our data to make conclusions about the consistency and sufficiency of system-generated IRC values. Each of the following sections describe these steps in detail.

5.1.2 Expected results and successful endstate

We expected that the IRCspec system would achieve results that are more consistent (smaller variance in mean differences between designer assessments) and accurate (closer to the objective design model established by expert consensus), compared to those results achieved using a manual method of estimating an IRC value. We also wanted to validate that interrater reliability was sufficient, and higher than reliability for the manual method. Such a finding would validate the IRCspec system as an adequate tool for articulating an abstract design model for a notification system. Additional flaws or shortcomings with IRCspec would be identified through general post-test questions about the comprehensibility of the specification tool.

5.2 Do Notification Designers *Really* Manage IRCs?

In order to refine understanding of the current design practices and needs of designers for a tool like IRCspec, we started out by analyzing designers' tendencies to manage IRC parameters in notification design, whether implicitly or explicitly. We suspected that experienced, professional designers and researchers carefully manage IRC-related tradeoffs in their design efforts, and that design rationale would often express these concerns and the resulting design decisions and research foci. However, we thought that novice designers would often lack such focus in their design rationale.

5.2.1 Method: Analysis of design rationale

To get an idea of how often professional and novice designers considered IRC-related tradeoffs in the design of a notification, we conducted an analysis of design rationale records.

Selection of design rationale records. For the professional efforts, we decided that it was important to consider specific interfaces as well as broader design themes or guidelines that related to notification systems. Since we had already developed a comparative collection of systems, we selected four of the systems depicted in Figure 4.4 to further analyze. In doing so, we choose to avoid the four systems already discussed in Section 4.3.1 and thus provide additional, more fully documented examples of IRC-related design rationale. The four specific notification system interfaces that we selected were: Scope [126], Sideshow [19], Kimera [78], and ambientROOM [68]. Selection of these systems was not entirely random–we knew that each were described in full papers from HCI conferences (CHI, CSCW, and AVI), so we thought that they would have the most fully developed records of design rationale. Another consideration was to have a sample of systems that can be classified in a variety of places throughout the IRC design space, and were implemented on various platforms (Kimera is a large screen display, ambientROOM is an augmented physical environment, and the other two are desktop systems, one adaptive and one nonadaptive).

To further explore tradeoffs of professionals, we also selected four papers published in HCI conferences (CHI and UIST) that described broader design themes for specific types of notification systems. Unlike the interface papers, these had not been previously read or analyzed in the context of notification systems design considerations. The papers were a result of a literature search on themes related to notification systems design: strategies for "invisible," ubiquitous computing (or *unremarkable computing*) [121], considerations for instant messengers [128], and information and interaction design of web-based advertisements (animation in online banners [14] and interactive advertising [105]). These papers were selected from a pool of eight theme papers because they seemed to address a wide range of design concerns, platforms, and IRC design models. We also ensured that they had sections that included specific design guidelines. As with the interface papers, screening was not based on inclusion or non-inclusion of IRC-related content–we wanted to characterize *typical* concerns of professional designers and researchers.

To contrast the professional efforts with novice efforts, we analyzed sample student design reports from three different groups:

- The first group of consisted of the most novice designers, many of whom had not yet completed an HCI course and were not at all familiar with our specific research area. The nine reports from this group came from a six-week long extra-curricular contest, in which each undergraduate team (often individuals) designed a solution for an airplane ticket price notification system (referred to as the *Expedia* problem). Contestants were motivated by $400, $200, and $100 prize offerings for "best usability" (additional contest guidelines are summarized in the notes for Table A.2); the design rationale was part of the submission.

- The second group of reports came from semester-long team design projects in an undergraduate HCI class. All projects required re-analysis and redesign of the Scope interface. Students had the benefit of reading the associated paper ([126]) and building from the original designers' rationale, but were not required to direct their efforts at any specific issues. Out of the fifteen teams in this class, five decided to radically change the interface (rather than tweak specific features of Scope)–these five reports were selected for analysis.

- The third group of student design reports came from the final project assignment in a graduate Usability Engineering class (a one-month long, five-person team effort at the end of the course). These students were required to design a notification system that would deliver CNN news alerts to users and entice them to access detailed content. Prototypes for all seven systems were evaluated by users in a single experiment, allowing the interfaces to be ranked in terms of closeness to a specific design model (criteria that guided the design efforts). We selected the reports from systems that were ranked highest (system A), lowest (system G), and in the middle (system C). Each report was the length of a typical full conference paper. Students were free to cite literature on notification design, and several were familiar with our research efforts.

Analysis procedure. The specific procedure used to characterize each design rationale artifact was straightforward. Using Table 4.2 as a guide for identifying interruption, reaction, and comprehension user goals, we highlighted mention of similar considerations in each paper. For all groups except the HCI Scope project, this task was intended to be comprehensive–every IRC-related phrase was noted. Since the HCI students were required to focus their reengineering efforts on mitigating one key problem claim for the original system, we noted any portions of the claim and the redesign proposal that related to IRC concerns. For all reports, we also made note of any other design theme that seemed to dominate the designer's thinking, such as customizability. Tables were constructed for each group of reports, depicting all phrases within the design rationale according to the IRC parameter that the phrase seems to address (see Tables A.1 and A.2).

5.2.2 Professional and novice designer results

In general, the exercise of explicitly listing instances of IRC-related rationale for professional design efforts reinforced our feeling that notification designers *really do* manage concerns related to IRC constructs. Students seemed to be much less consistent in their coverage of IRC tradeoffs, although they seemed to be able to pick up IRC themes after careful examination of professional design rationale products (i.e., the Scope paper). When students did discuss IRC goals, the concerns cited often seemed to lack insight, except in the best cases.

Professional papers. Each of the eight professional papers included multiple phrases that pertained to each of the three IRC parameters. For example, all designers and researchers reported some consideration of interruption, although terminology and specific goals differed. Many referred to "minimizing distraction" or "safeguarding attention," although others intended to elevate noticability, support "variable distraction," and prevent users from ignoring notifications. Almost all papers expressed deep consideration for goals that we summarize with the IRC parameters, frequently treating them as tradeoffs and expressing specific levels desired. As seen in Table A.1, each parameter usually had four or more associated phrases in the design rationale, although designers did discuss other important design themes as well (i.e., goals for adaptability, poetic representation, and acceptability in domestic settings).

As critical parameters should be, the IRC parameters within the design rationale can certainly be thought of as points of comparison between the interfaces and design guidelines. For instance, several designers suggest an intention for low interruption (System 1 - 5), while others expressed an intention for higher, situationally demanded interruption (System 6 - 8). When intended levels of the other two parameters are also considered, we improve the ability to distinguish the design intentions from each other: high reaction is a goal for Systems 1, 2, 4, 7, and 8 (but not for the others); high comprehension is targeted for only Systems 1, 2, 4, and 6. Thus, the most comparable sets of systems in terms of design intentions are Systems 1, 2, and 4, Systems 3 and 5, and Systems 7 and 8. If a new designer wanted to see an example of a system designed to target high interruption, low reaction, and high comprehension, they should refer primarily to System 6.

Student design projects. In general, the students were much less thorough in documenting any IRC-related concerns they may have considered in their design. There were key differences in the three groups of students, and notable exceptions that hint at broader potential for improvement.

The first group (designers of the Expedia notification systems) seemed to have major problems expressing many insightful design rationale points that related to IRC concerns, even though they were specifically instructed and motivated to do so. Only two out of nine reports showed any consideration of all three parameters, and only one other report showed very cursory consideration of more than one parameter. Even the most thorough rationale (System D) expressed few propositions that related to user goals for the critical parameters. Certainly, this result can be attributed to several factors: the length of rationale statements provided was generally only a few sentences, and students had no formal training on how to write design rationale or think about psychological effects of situation features or artifacts in use. However, it is arguable these may be typical characteristics of many novice designers, and part of the difficulty in this task was decomposing and articulating concepts related to interruption, reaction, and comprehension.

The second group (redesigners of Scope interfaces) generally provided more encouraging examples of design rationale that was intended to address IRC user goals. Students seemed to be making critical claims about the original Scope interface that addressed problems in supporting IRC-related user goals. Frequent comments included the ideas that Scope is too cluttered to be glanceable, takes up too much screen space to be non-distracting, fails to support desired user reactions (discerning and deleting alerts and high urgency items), and interpreting the notifications being monitored. Perhaps much of the students' success in picking out IRC-related concerns must be attributed to their careful reflection on the Scope paper and its focus on IRC concerns. It was somewhat disappointing that only two of the critical redesign claims seemed to address IRC-related goals that the original designers did not think of (System G, (R): awareness of time in which a notification should be handled should be factored into the display of items; System H, (I): persistent display of to-do items can cause user stress, and thus, interruption). This suggests that a broader appreciation of IRC constructs might have allowed students to recognize more insightful problems.

Results from the third group show the highest variation. The team that designed System A, the most highly rated CNN notification interface in the usability evaluation, had the most well

Group	*Experience Characterization*	*Instances of IRC-related Design Rationale*	*Complexity of Issues Addressed*	*Full Record Reference*
Professional designers or researchers	Published in main HCI conference; part of prominent university or corporate research center	Frequent, full coverage	High	A.1
Expedia Design Contestants	Undergraduate computer science majors, pre-HCI	Incomplete, cursory	Very low	A.2(1)
Scope Re-designers	Undergraduate computer science majors, taking HCI, familiar with at least one professional notification system design effort	Frequent	Moderate, but few original	A.2(2)
CNN Notification Designers	Graduate Usability Engineering class	Full range: frequent to cursory	High variation	A.2(3)

Figure 5.1: Summary of the four groups (one professional and three novice) from which design rationale of designed notification systems was analyzed. The results from the Expedia contestants and the CNN Notification designers were most concerning, as the design rationale consistently lacked focus on key psychological effects of notification.

developed design rationale, in terms of IRC goal coverage. However, at least one member of this group had previously attended seminars that discussed our notion of the IRC design space and was familiar with associated literature. Team G's design rationale addressed all three critical parameters, but seems to be much more simplistic than the other two groups. Their interface scored the lowest in the usability test, especially in assessments of "annoyance factor" and supporting "response to content." For this group of students, ability to make explicit IRC-related design rationale seemed to correlate with subjective assessments of their interface usability.

Key findings. In publication of their work and design rationale, professional designers of notification systems tend to place some emphasis on describing intentions for all three critical parameters. Not only is this evidence that good designs explicitly manage the critical psychological effects, but we can note the diversity of intentions for the critical parameters within even this small sample of systems. Certainly, if we are to compare designs or reuse design components, we ought to have at least a high-level appreciation for the original designer's intentions (e.g., "low" vs"high" parameter levels). The survey of professional design rationale provides encouragement that the IRC parameters can form a common framework for understanding notification system design intentions.

In general, the tables summarizing student design rationale records provide evidence that novice designers may have difficulty creating insightful IRC-related design rationale. When students studied examples of professional design rationale or were knowledgable about the IRC framework, they were able to recognize and attempt to support IRC goals in their design efforts. This reinforces and channels our motivation for providing a tool that would help designers consider and characterize IRC-related requirements in notification systems design.

While we found that professional designers *really* did consider IRC-related tradeoffs in design, this did not seem to be innate quality of all novice designers. The next section discusses how we transferred many of the observations made in this study to the design of such a tool–IRCspec.

5.3 IRCspec Tool Design and Tuning Process

Our analysis of design rationale produced several ideas that guided the tool design. First, we discuss the root concept which emerged, and high-level activity design decisions that followed. Next, we review the process of selecting and organizing specific content to guide consideration of IRC parameters in notification systems design. Finally, we discuss the iterative process used to formatively test and "tune" the IRCspec tool so that it might produce the most consistent and accurate results. The next section (5.4) describes the validation efforts for the tool.

5.3.1 Tool design considerations

High-level activity design. From our own requirements analysis, we learned that novice designers tend to have difficulty with considering and expressing tradeoffs related to critical parameters in notification systems design, or IRC. We speculate that supporting this process will help novice designers probe key questions routinely addressed by more experienced designers. Helping more designers to think in terms of a common design space will catalyze growth of a variety of potentially reusable design solutions. Thus, the root concept for the IRCspec tool is *to expose novice designers to essential tradeoffs of typical notification system design goals, and to help characterize their design intentions accordingly.*

In doing so, we anticipate that several general assumptions and user activities will be important:

- Using IRCspec should not require any specialized knowledge about human information processing, the dual-task paradigm, or the IRC framework.
- Although the tool should prompt designers to consider facets of the multitasking experience, it is likely that many of these facets will be difficult to anticipate in the requirements analysis stage.
- The system should help draw designer focus away from platform-specific tradeoffs and constraints. Since notification systems can be implemented on many different platforms and

involve a variety of information design strategies, it will be especially challenging to channel concern to the information need and situational context associated with the notification delivery.

- The process of using the tool should be a simple, brief transaction, allowing designers to gain proficiency with repeated use, to the point that the tool is soon unnecessary.

With these goals in mind, we started brainstorming about what the process of using IRCspec could be like. In order to generate as many options as possible, we made this brainstorming an exercise in three different seminars (each consisting of about eight students that had been spent several previous weeks learning about the IRC framework and contrasting various notification systems). Students were asked to develop an approach they thought might work, demonstrate its use, and reflect on the approaches taken by others. We were able to identify several possible options by observing these discussions (see Figure 5.2).

Several different rough prototypes were developed and discussed to explore each approach. Students reverse-engineered requirements for a few different example systems, and used several different prototypes to obtain design model IRCs. After comparing consistency and satisfaction with the results, we still were not satisfied with any of the approaches. The guideline approach proved difficult to learn and apply consistently, the additive approach seemed to be quite difficult to construct without platform-specific design feature references and it was also difficult decide the order in which parameters should be considered. Finally, the categorical approach seemed require so much supporting material that it would scale poorly for the whole design space.

However, two key points emerging from these initial studies motivated our eventual approach.

1. First, the additive approaches certainly were the simplest to use, and were compelling under the goal of promoting learning. However, we started to realize that composite parameter factors could not just be simply described as potential parts to be added, independent of the other parts. Instead, we started to envision each parameter as a complex and generic concept that included several components which might be weighed differently in various combinations.

2. In assessing tool designs that would be most effective for novice users, we saw clear benefits in prototypes that presented minimal chunks of information at any time and helped users recognize possible choices. A wizard-style interface that presented direct questions came to mind.

Still largely uncertain about how the parameter selection or calculation would take place, we started to explore the specifics of what a wizard-style questioning tool would need to contain. As we continued, we wanted the tool to be a metaphor for the designer writing a story about the user's notification goals in a dual-task situation.

	Categorical	**Additive**	**Guidelines**
Overview	*Describes various levels of a given parameter, in which a designer would find the best matching attributes by considering tables or a decision tree.*	*Factors within a checklist are totaled up for each parameter rating.*	*General statements that could be used to think about critical concerns within each parameter.*
Example	*Interruption:* 1 – .8 pop-ups, large amounts of screen space required .8 – .6 flashy animation or audible chime intended to distract typical user ...	*Notification introduction:* +.3 I any audio +.2 I multiple animation schemes +.1 C icons and tooltips +.5 C new item related to larger trends *Monitoring status:* −.2 I subtle changes +.4 R easy-click buttons	• Impact on senses – While most notifications systems employ visual or audio interruption… IRC should be rated relative to the impact on the senses. • Intrusion into primary task – The other important aspect of interruption is …
Upsides	+ fairly easy to use if descriptive enough	+ easiest to use if organized by typical interface transitions	+ offers maximum flexibility and designer discretion + easy to support from literature
Downsides	− constrains early design thinking to specific platforms − requires exhaustive list of design strategies/artifacts − validity would be difficult to establish	− constrains early design thinking to specific platforms − very difficult to create plausible scoring system	− numerical rating process still very subjective − different guidelines may be required for different parts of the design space

Figure 5.2: Three alternate approaches to generating a design model IRC (categorical, additive, or guideline based) that were initially considered. All three approaches were discarded after prototype methods failed to suggest consistent selection of choices, feasible scalability for an entire design space, and for other reasons discussed in Section 5.3.1. Instead, the IRCspec tool was developed to implement the concepts depicted in Figure 5.3 and the algorithm described in Appendix A.2.

Selection and flow of questions. To summarize our strategy for developing IRCspec: we decomposed the dual-task situation into key concerns, articulated the concerns and a general test for notification system characteristics through simple questions and potential options, and designed a system that could present these questions to a user.

We began with the excellent discussion of dual-task situations provided by Wickens and Hollands [135] (our overview this and resulting taxonomy is included in Section 4.1). From this and the concerns discussed by experts in our design rationale study, as well as literature specifically discussing notification system design challenges [90], we identified several important parts of the dual-task situation that should be considered when designing a notification system. Key concerns include:

1. desired effects (user goals) of receiving a notification,
2. the relative importance and relationship between tasks,
3. the relationship between the notification and existing knowledge,
4. monitoring and interpretation effort required due to characteristics of the notification (e.g., complexity and frequency).

With the intention of identifying questioning steps within a wizard interface, we developed a flowchart of these key concerns. Several assumptions guided this effort. First, we anticipated that a single list of questions would not allow sufficient clarity, since several consideration items and options seem to depend on other variables in the situation. For example, the delivery frequency of the notifications will be important to consider if the delivery is known to be regular rather than data-driven. This consideration led to some conditional flow points in the series of questions, particular to clarify questions that may be confusing. Second, based on reviewer concerns noted in Section 4.5, we wanted to ensure that the system being designed is being conceived as a notification system (as opposed to a more general information system). Therefore, a brief series of ideas was included to check whether the system would be typically used in a dual-task situation and outside the user's primary or usual attentional focus. Finally, we wanted to probe the IRC-related goal structure both directly and indirectly. With the thought that the earlier questions relating to the key concerns would provide indirect probes and stimulate a designer's thinking, we included steps at the end of the process to directly query the levels of welcomed interruption, expected reaction, and desired comprehension. The basic flowchart for this system is depicted in Figure 5.3.

Next, each of these concerns was phrased as a question, with multiple questions for more complex ideas. All questions were phrased in future tense, reflecting the intention that IRCspec be used to anticipate user goals and set a design model prior to design of the notification system interface. Since the process is intended to be straightforward and not assume any previous knowledge about designing for dual-task situations or the IRC framework, all questions are written in language that is as simple as possible, and the number of questions was kept to a minimum (no

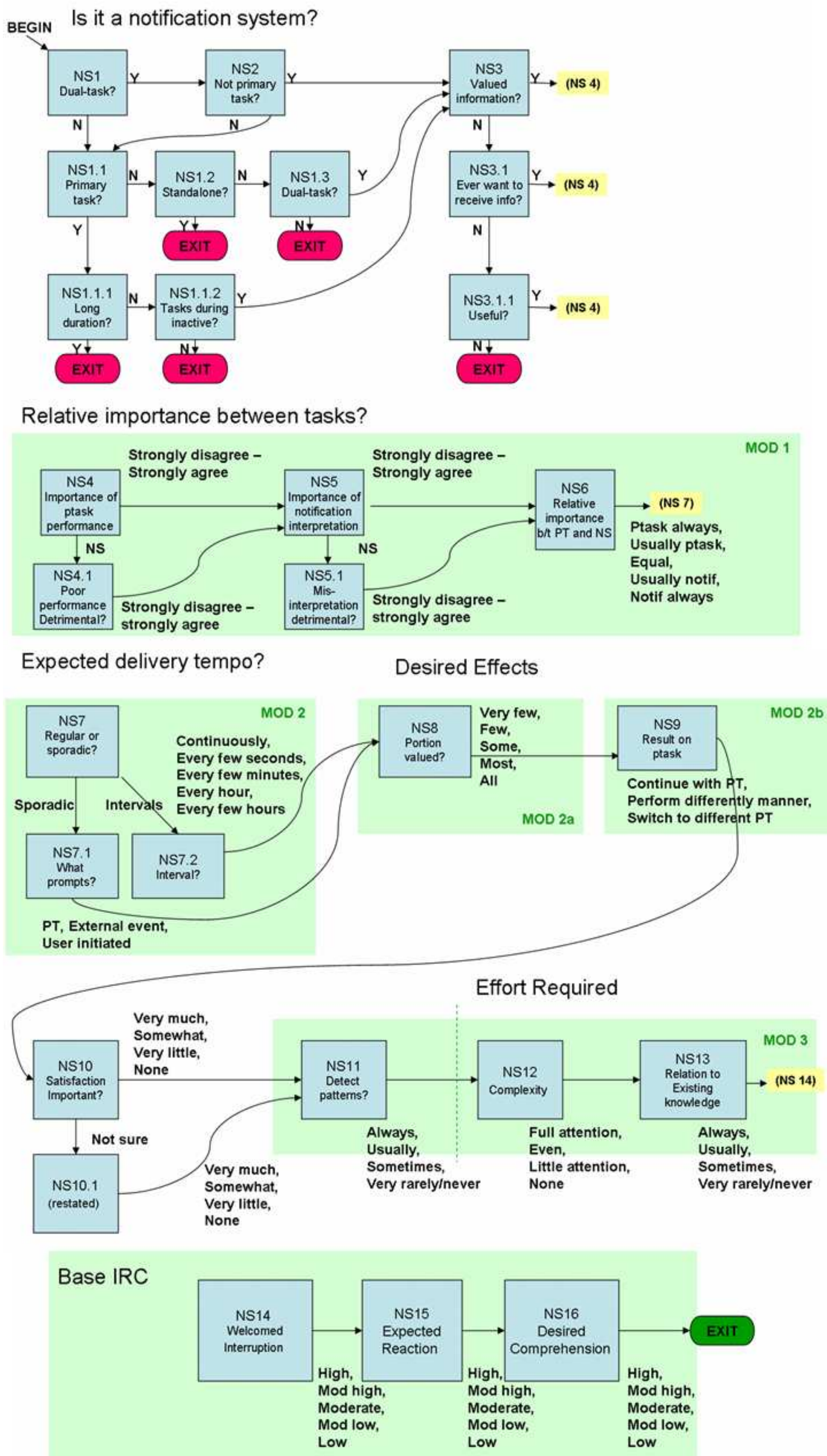


Figure 5.3: Flow chart of the IRCspec system.

more than 20). Question writing was a result of a semester-long team effort that included two undergraduates [1] and was guided by iterative review by others familiar with dual-task design. Actual questions and answers can be found in the appendix, Section A.2.

After the questions and answer choices were established, we reasoned about the effect each possibility would have on one or more of the parameters. To do this, we first identified the operative parameter or parameters and then set the implied range of values. For instance, one of the selections might imply that C should be “greater than 0.75,” while another selection might imply “less than or equal to 0.3.” Although this process was somewhat subjective, we applied it consistently for all choices and used expert walkthroughs to tune the values, as described in the next section. The basic idea of this initial scoring algorithm was that the parameter ranges would be narrowed throughout the questioning process, until the direct question relating to each parameter that established a “base value” for each parameter. At that point, the range endpoints and the base would be checked for consistency and averaged appropriately for a final value (see Section A.2 in the appendix for additional details).

This initial algorithm led to the creation of the IRC equations that are introduced and discussed in the next chapter. To simplify the IRCspec tool and increase its compatibility with the user’s model assessment tool(s) developed later, the underlying algorithm was later replaced with the equations (answer choices set the equation variables). However, because the initial algorithm was an essential part of the tool’s evolution, we report the formative testing process.

5.3.2 Tuning and retuning the underlying parameter values

With the questions, question flow, and underlying algorithm developed, we began tuning the quantitative weightings embedded in a prototype of the tool. Methodologically, there was more than one way that we could approach this step. Certainly, one standard approach would be to obtain scaled, numerical responses from a group of experts for a full range of systems, and then use this data to set or adjust the quantitative weightings for each question and for overall parameters values. Factor analysis and factor loading would be a viable option with this approach, allowing identification of specific questions that should be assigned more value in certain conditions. An alternate approach was to use a few test cases to validate quantitative weightings set through consensus of multiple researchers. While one might argue that the first approach might deliver a more statistically reliable tool, resources required for complete application of this method far outweighed the need for an initial indication that the tool might work well enough to be used in a more summative test. Certainly, the factor analysis approach should be used to fine-tune the tool or prepare it for wide-scale use. For the purpose of tuning and pilot-testing, we had several experts[2] (not involved with the initial setting of quantitative weightings) consider a few test cases.

Each trial involved one of four notification systems–systems selected to be representative of the complete design space. Each of three experts that assisted with the process was very knowledgeable about the IRC framework and the system or systems they were asked to assess. We began

[1] Ben Devane and Steve Battjer

[2] Jacob Somervell, John Booker, Shahtab Wahid

the trial by describing a general interaction scenario for the notification system under analysis (although the Info Art interface was used in two different trials, the interaction scenario was quite different in each). Experts were then asked to individually reverse-engineer and note an estimated IRC design model for that system. Then, they each used the IRCspec prototype to assess the design model with our questions and underlying algorithm. The initial estimate ("Est. IRC") and calculated values ("Final IRC") for each trial are shown in the chart found in Section A.3. Also depicted here is the verification process that the algorithm was functioning as designed, plots of the adjustment ranges and the possible (base) ranges for each parameter.

Following each trial, we received feedback from each expert regarding the clarity of the questions and choices, as well as their satisfaction with the match between their estimated value and calculated value. Changes to the prototype were made iteratively after testing each expert. For example, P3 (who was actually the first expert) had a large mismatch (.4) between R-values. After discussing this with the expert, we learned that the lower limit on the MOD-2b R-value was probably set too low. We were encouraged that the last expert we tested (P1) had a no more than a 0.15 difference between any estimated and calculated IRC value, although only a 0.5 difference in four of six parameters for the two systems assessed.

Having successfully conducted a walkthrough with experts, we continued development and testing of the tool as described in the next two sections. However, as mentioned earlier, we eventually replaced the algorithm with equations that modeled the general behavior more simply and descriptively. Since the questions had not changed and we did not want the expert-derived IRC values to change by more than an average of 0.1 point per parameter, we used the original responses to drive a simulation with the equations. This helped tune the variables newly associated with each question choice.

5.3.3 Putting it all together: The IRCspec tool and system integration

With the initial prototype and scoring method satisfactorily tested by experts, we continued development of the IRCspec system by providing a robust, interactive user interface. In keeping with our decision for a wizard-like interaction, we implemented a the series of questioning screens in Macromedia Flash and the scoring algorithm in Flash's ActionScript (see A.2). Users of the tool indicate answers using radio buttons, and screens are advanced using an "OK" button. At the end of the question series, the IRC values are calculated and provided. To allow users to gain knowledge about IRC scoring, encourage experimentation with decision-making, and eventually remove the designer's need for the tool, we included a "Previous" button. Answers can be changed for any of the questions and the IRC values can be recalculated to see the impact of any answer on overall parameter values.

The benefits of implementing IRCspec in Flash became apparent as we developed plans for a larger system, LINK-UP (discussed in Chapter 7), which would support a development process for notification systems. Using the web-based LINK-UP system, a user could obtain design model IRC by launching the IRCspec tool and having it return the values to the original webpage with Javascript commands. However, the Flash platform also supported standalone testing and use, to include writing results to an external file. The current version of IRCspec can be found at: http://research.cs.vt.edu/ns/IRCspec.exe. The next section describes the formal, lab-based test used to validate IRCspec's consistency and accuracy.

5.4 Validation of IRCspec Consistency and Accuracy

Having created a system that allows designers to reason about important dual-task factors and automatically generate a design model IRC, we wanted to establish the tool's consistency and accuracy. In order for the IRCspec system to be a thought of as reliable tool for expressing a design model, there must be results that indicate sufficient consistency and accuracy in matching a given design conception. Before the development of this system, design model IRCs were only roughly estimated by experts based on loosely defined concepts and example system classifications. The IRCspec tool should help designers of all experience levels with a wide variety of notification system models, as design conceptions can occur in all regions of the IRC framework and incorporate diverse information and platform needs of users.

5.4.1 Hypothesis, method, and procedures

Hypothesis. Specifically, we hypothesized that a user test with our tool would validate several system objectives. Our first objective enforces *accuracy of critical parameter establishment against expert consensus;* we expect agreement within 19%. This value is less than the average expert-to-expert parameter assessment agreement rates obtained in pre-testing with manual assessment methods (as explained in Section 5.4.2). Our second objective ensures *consistency–that different designers are able to derive similar critical parameter values given an identical design problem,* for which we also expect agreement within 19% and a suitable score for interrater reliability. These objectives apply throughout the full range of possible parameter values. Of course, we also expect that designers generating critical parameter values with this tool will obtain more accurate and precise results than designers with no tool at all (using manual, heuristic-based estimation).

Method and Procedures. We conducted a lab-based, single-factor, between subjects test with two treatments: 1) using IRCspec and 2) using the manual method to assess design model IRCs. Twenty participants motivated by class credit were included in the study, 10 per condition. All participants were computer science majors (sophomores), although none had taken Introduction to Human-Computer Interaction yet. We selected this population to minimize any previous knowledge about notification systems design, but still validate the system with targeted users (novice interface designers).

Before beginning formal testing, we tuned the algorithm with a number of system and requirements walkthroughs by different experts, ensuring expert users could achieve agreement between manual and tooled parameter assessment (as described in the previous section). The experiment was designed to be conducted within a 30 session, to include the informed consent process, experiment procedures, and administration. Small groups (3-5) of participants within the same treatment were tested together, although there was no interaction between participants and all groups received the same instructions.

At the beginning of the session, the participants were asked to think of themselves as a notification systems designer. They were provided with a definition of notification systems, as well as a few example systems. Participants were instructed that they would be answering a few questions about four different design problems, one a time. Each design problem was introduced as a single slide on a large screen display (the four design problems can be found in the appendix, Section A.4.1), read aloud by the experiment proctor. At this point, participants assessed the design model IRC using one of the two methods.

- Participants in the IRCspec treatment used a desktop computer with IRCspec installed, progressing through the wizard interface and recording their answers to approximately 16 multiple-choice questions. An example question is *"Which statement describes the general relationship between the importance of the primary task and receiving the notification?"*; all questions can be found in the appendix, Section A.2. After answering all questions, the parameter values were calculated with the underlying algorithm, and all input was stored in a log file. The one important difference in the version of IRCspec used for this test was that the calculated IRC values were not displayed at the end–participants actually had no awareness that they were being generated by the system.

- Participants in the manual treatment were provided with a form that helped them determine a design model IRC using the parameter definitions, a brief list of points to consider, and a few other basic instructions (Figure A.7 shows this form). They used this form to come up with the IRC value for each design problem and then recorded their assessment on a separate page (see Figure A.9).

Following their assessment of the design model IRC, all participants were then shown a series of possible notification interface designs and descriptions (also found in Section A.4.1)–they were told to select the most appropriate starting point for the problem specification and design model considerations. The system selections were recorded on forms. After the participants completed the four design problems, they were asked to fill out a post-test survey to determine if the questions addressed all factors they felt impacted interruption, reaction, and comprehension.

Prior to testing the novice designers, we obtained benchmark parameter values for each of the four design problems from two impartial experts[3] that assisted in the development of the IRC system (but not the IRCspec tool). Both experts first provided an IRC value derived through their own estimation and then used the IRCspec tool to obtain a second IRC value.

[3] Ali Ndiwalana, Scott McCrickard

5.4.2 Analysis and results

Expert benchmark analysis. We begin the discussion of the data analysis with the IRC values obtained from the two experts prior to the test with the novice designers. Figure A.10 shows the IRC values provided by both experts using the two methods for each of the four design problems.

First, we wanted to ensure that there was a reasonable level of agreement between an expert's IRC calculations with the two methods. That is, an expert should be able to obtain similar IRC values for a given design problem regardless of whether they used a manual estimation method or the IRCspec tool. To check this, we simply calculated the absolute difference between each method for each parameter. For example, Expert 1 manually estimated Design Problem 1 to require an IRC value .4/.4/.7 (the three numbers refer to I, R, and C, respectively), but obtained .5/.5/.7 with the IRCspec tool. This reflects an absolute difference of .1/.1/0. Differences for each design problems and both of the experts are included in Figure A.10. Expert 1 had an average difference of .10/.05/.10, without any parameter in any of the design problems having a difference of more than .2. Expert 2's average difference was higher: .25/.25/.20, with at least one parameter in each of the design problems exceeding a difference of .2. This analysis implies:

- All four of the design problems *could* produce consistent results between the two methods for each parameter; therefore, they were suitable for use in the novice testing.
- Expert 1's results would be most appropriate to adopt as benchmark values for the testing of the accuracy portion of the hypothesis, since his were the most consistent IRC calculations. This option was thought to be better than averaging the two expert result sets or using a set of values established by the researcher that wrote the design problems and developed the IRCspec algorithm.

Next, we were interested in which method provided the most consistent results between the two experts, so we simply calculated the absolute difference between experts for each parameter. For example, in the first design problem Expert 1 manually estimated the IRC values to be .4/.4/.7, while Expert 2's estimate was .2/.1/.8. This reflects an absolute difference of .2/.3/.1. Differences for each of the methods and design problems are also included in Figure A.10. We found it encouraging that, between the two experts, the average difference for the four design problems was less using the IRCspec method than the manual method (IRCspec was .10/.10/.10, manual was .20/.15/.23). *Determining the average manual difference between experts (slightly over .19) established the target value for testing the IRCspec accuracy with the expert benchmark and consistency between novices*–we would not consider the tool to be successful unless the average difference and standard deviation was lower than .19.

This finding is echoed in an analysis of Kendall's correlation of interrater reliability between the two experts (see Figure A.11 for details). That is, we examined the correlation of the two expert ratings for a each parameter (I, R, and C) and design problem, expressed by Kendall's τ. *Interrater reliability was significant for the IRCspec results (*$\tau = .59$*,* $p = .013$*), but not for the manual method (*$\tau = .45$*,* $p > .05$*).* While comparing the reliability between two expert result sets supports the portion of our hypothesis related to consistency, we recognize that the sample size is too small for any conclusive inference.

Novice accuracy analysis. The second phase of the data analysis involved making inferences about the accuracy of the novice designer design model IRCs, with respect to the benchmark values calculated by Expert 1. Since accuracy is simply the measure of conformity to a model, we began this analysis by determining whether the novice participant ratings fell within .15 of the expert benchmark calculated with the same method. For example, Participant 1 (who used the manual method) determined a design model of .7/.5/.5 for Design Problem 1. Expert 1's manual design model IRC was .4/.4/.7. While the R-value was within .15, the other two parameters were greater than .15. At the same time, we determined whether ratings were more than .30 different than the expert rating. Figure A.12 shows the raw data for all participants, shaded according to expert agreement level. To determine preliminary results, we counted instances by method where a high or low level of agreement with experts was observed. For both categories, *the IRCspec method was more promising than the manual method: there were 21% more ratings within .15 of expert ratings and 12% less ratings more than .30 away.*

The next step began with a calculation of the exact absolute difference between each novice designer IRC rating and the corresponding rating by the expert (see Figure A.13). Again, the descriptive statistics were encouraging. *The overall average difference for the IRCspec tool was .16 (satisfactory), but .25 for the manual method.* With a satisfactory overall average difference, the first portion of our hypothesis was supported–IRCspec helped novice designers produce IRC values that were accurate, compared to those generated by Expert 1. Participants using IRCspec also seemed to have more consistent areas of expert disagreement, hinting that the next phase of the analysis may be fruitful as well.

This analytical approach was continued to determine whether the two methods were actually different in the support they provided for accuracy. Since we are considering the design model data generated by either method to be ordinal, a parametric test like an analysis of variance (ANOVA) might not be accurate. Therefore, we used the Wilcoxon Rank-Sum Test [5][4]. In this non-parametric testing procedure, we rank-ordered the differences produced by each method (in a common pool), assigned ties the average rank, and then found the sum of the ranks for each of the two methods. The observed rank-sum for the manual method, w_{MA}, was compared to the distribution of the rank-sum, W_{MA}. Our hypothesis (H_1) was:

The distribution of dispersion would for the manual method would be shifted to the left (or upper-tail), indicating larger differences from the expert ratings.

Although the distributions of the two samples were slightly skewed to the left (see Figure 5.4), the distribution-free transformation inherent in this non-parametric test prevents this from being an issue. H_1 was supported by a one-tailed test on the rank-ordered data, $p = 2pr(W_{MA} \geq 15700.5) = .001$, validating that *IRCspec provided greater accuracy in IRC values than the manual*

[4]See http://www.stat.auckland.ac.nz/ wild/ChanceEnc/Ch10.wilcoxon.pdf for a guide to the specific procedures

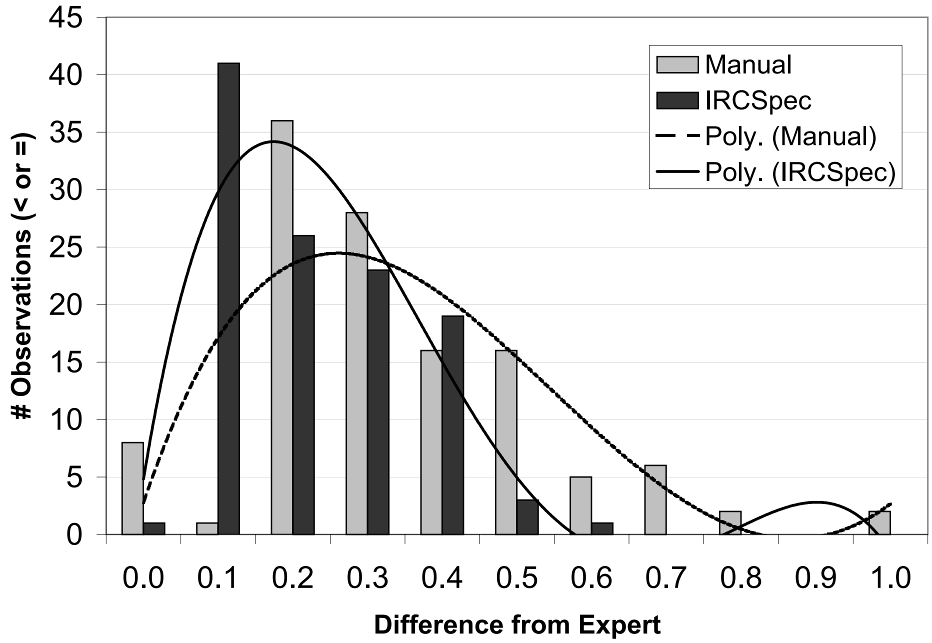


Figure 5.4: Distributions of the design model parameters samples collected using the IRCspec method and the manual method. Polynomial trendlines (4th order) provide a sense of the distribution shape.

method.

Novice consistency analysis. The third phase of the data analysis probed the consistency of the participant ratings for each of the methods. We approached this in two ways: the consistency of the raw ratings (ignoring agreement with the expert benchmark) and consistency of the absolute differences from the expert benchmark. The first approach was important for recognizing portions of the test where the tool resulted in wide differences between users. For this line of analysis, we continued to work with the raw data in Figure A.12. The second approach indicated the actual reliability of the tool to serve its intended purpose–allow any novice designer to obtain a design model IRC that is similar to one an expert would specify. Figure A.13 was the base of this line of analysis.

In examining the consistency between the raw ratings, we were encouraged by the lower standard deviation averages between participants using the IRCspec method to calculate a given parameter, as seen in tabular form in Figure A.12 and graphically in Figure A.14. For half of the parameters calculated with the manual method, the standard deviation was greater than .25, implying that even a "high" or "low" agreement might not be reliable. However, *only one of twelve parameters calculated with the IRCspec tool resulted in a standard deviation greater than .20, and the average was .14.* We further analyzed the raw scores with a test of interrater reliability for each design problem using Cronbach's coefficient α. The intermediate values (variance for each parameter and total ratings for each design model) and results can be found in Figure A.15.

The overall average of all four design problems demonstrates sufficient interrater reliability (.69) of participants using the IRCspec tool, however, the applicability of this test is questionable due to the meaninglessness of summing the three parameter ratings to obtain the total score.

With the second approach, we continued to probe the absolute differences discussed earlier and depicted in Figure A.13. We determined the interrater reliability between participants using each tool by finding the variance in all raw differences for each parameter, as well as the sum of differences for each design model (in this case, a meaningful summation operation). At that point, we determined Cronbach's coefficient α for both methods, finding that *IRCspec had an acceptable interrater reliability value (.67) while the manual method did not* (see Figure A.16). Considered with the acceptable standard deviation of participant ratings using the IRCspec tool, results support the hypothesis that consistency between novice users will be strong.

5.4.3 Study implications

Our lab-based study supports all facets of our hypothesis–IRCspec allows different designers to derive similar critical parameter values given an identical design problem, values that are sufficiently close to those that would be obtained by an expert. Users of the IRCspec will calculate more accuracte design model IRCs than novices that use a manual estimation technique.

Claim 4 – The IRCspec tool…

Analysis and Results, **Section 5.4.2**

+ provides high interrater reliability between experts

+ allows novices to more frequently achieve IRC ratings that are close to expert ratings

+ provides greater accuracy, with respect to expert benchmarks, than a manual estimation method

+ results in more consistent IRC ratings and higher interater reliability between novices

– may be tedious and unnecessary for expert users

In a usability engineering process, the first step is generally gathering and analyzing user requirements to drive interface design, to include understanding tasks, information characteristics, user background, and other aspects of the situation. In Norman's terms, this forms the design model, based on dimensions of successful dual-task design recognized in research. We observed difficulties in this step in our analysis of design rationale. With IRCspec, notification systems designers of all experience levels are provided with convenient access to these considerations. The system ascertains the critical parameter levels of desirable user interruption, reaction, and comprehension (IRC values) expressed simply as triplet of ordinal scale values between 0 and 1.

We also observed several behavioral characteristics of tool users. Novice participants tended to concentrate very hard on the questions, frequently going back and forth between questions and sometimes changing answers. This was exactly the type of behavior we hoped for, reinforcing our choice of a target user group. However, since the participants did not see the IRC value calculated by the IRCspec tool, we are unable to make any guesses about the tendency of the tool to support learning of the IRC framework. This is an important direction for future work, as we also informally observed that expert users are more apt to continue to make unstructured estimations of a design model IRC, rather than access and use the tool.

Our ability to successfully support accurate and consistent calculation of notification system design models has broad implications, which have been described in our published work [32]. If IRCspec were to be integrated with a claims library or a more complete design environment (as discussed in Chapter 7), designers could search for influential and reusable claims from previous projects based on critical parameter requirements. Design knowledge could be gathered in a manner similar to the Internet shopping cart metaphor used on e-commerce sites. While the IRC values might be the most influential index, other indices would also be used to access this design knowledge, to include the generic tasks that the system will support (e.g., monitoring or alerting), design choices (e.g., use of color or animation). Much of this other information can be gathered from ETAG (Enhanced Task-Action Grammar) specification [43] or direct input by the designer.

As we consider other broad implications resulting from the success in developing and validating the IRCspec module, we recognize that we have taken an important step in overcoming a key challenge in the use of critical parameters to organize and access design knowledge. However, we this reminds us of the other challenges, which are introduced in the next section.

5.5 Conclusions and Future Work

This experience of developing IRCspec helped recognize several general challenges with using critical parameters to guide design knowledge reuse (as we propose to be a broader implication). We summarize each challenge, commenting when appropriate on how it was addressed in this research.

- **Target appraisal.** Designers must be able to transform abstract requirement variables to qualitative critical parameters. Although requirement variables for any class of system (which describe the design model) are likely to be quite numerous with wide ranges of possible values, some mechanism must be present that funnels these variables into abstract design goals expressed as critical parameter values. This is the specific focus of IRCspec.

- **Iterative assessment.** Designers must be able to estimate critical parameter values throughout the design cycle to gauge the impact of decision-making on design progress. In short, analytical and empirical testing processes must be able to calculate effects necessary to determine whether the critical parameters will be reached. The next chapter will address this challenge.

- **Benchmarking.** Through iterative assessment, benchmarks must be established to summarize state-of-the-art effects of actual systems used in real world situations. In this case, design characteristics for specific parameter ranges (e.g. low interruption) would be collected, assisting other designers with understanding implication of various parameter values. This is also a challenge noted by others, which can be used to form reference tasks for research programs [133]. A benefit of an automated system like the one proposed in the previous section, is acceleration of consensus and collection of benchmarking data–an idea further explored in Chapter 7.

- **Definition.** A common conception of parameter definitions, as well as acceptable units and methods of measure, must be established so that they can be universally applied–a process worked out through the acceptance of benchmarks. While the researchers may be moving toward common definitions of essential usability metrics, there is still a long way to go. Certainly, related work in the behavioral science fields provides a good starting point that can be bridged to the specific needs of interactive design.

- **Selection.** Researchers must be satisfied (and satisfy others) that they have exhaustively included the right parameters in consideration of the system class and that all parameters apply to all systems within that class. As our research area matures, acceptance of these parameters must become more widespread.

This effort provided a tool, IRCspec, that allows consistent and accurate specification of design model IRC parameters during requirements analysis for a notification system. As discussion of this research continues in the chapters that follow, we describe other instances where IRCspec was used and assessed in the context of real design efforts, rather than a simulated lab-based experiment. However, the success in deliberately analyzing the need for this tool and validating its key system features in a controlled study has provided confidence that the other challenges with using critical parameters can be overcome.

Chapter 6

OBTAINING USER'S MODEL IRCs FROM USABILITY TEST RESULTS

This chapter addresses the need to enable comparison of design and user's model IRCs—a precursor for improving generalizability and reuse of design knowledge. As the second of three key aspects of operationalizing the IRC framework for notification systems design analysis, this step builds on the efforts reported in Chapter 5 that involved developing the IRCspec tool. Most of the material presented in this chapter was included in a full paper [38] appearing in the ACM Conference on Designing Interactive Systems (DIS), August 1-4, 2004.

Section 6.1 presents the underlying motivation, problem, and expectations of the project. To extend the conclusions from the IRCspec project and to frame the requirements for gauging user's model IRCs, we present an analysis of issues with using critical parameters to guide a design process in Section 6.2. In an effort to mitigate these key concerns and provide a mechanism for obtaining user's model IRC values, we created a generic evaluation tool, referred to as "IRCresults" and introduced in Section 6.3. This chapter also includes two lab-based case studies that illustrate the use of IRCresults. The first case (Section 6.4) validates that consistent, yet expressive analytical usability conclusions can be obtained with IRCresults, and also includes analysis of the design-user's model disparities. The second case (Section 6.5, which was not included in [38]) demonstrates a pilot-test of a lab-based, empirical usability evaluation with IRCresults.

6.1 IRCresults Motivation

We have provided initial examples of notification system and design artifact IRC classifications, as well as a demonstration of how IRC parameters could guide a walkthrough of a human information processing model [89]. Exploratory work probed the use of IRC parameters for indexing mechanisms to notification systems design knowledge repositories [101], and identified general challenges with using critical parameters in systems supporting design knowledge reuse [32]. We pursue a long-term vision of enabling integrated *claims reuse* in a software design process, a proposal advocated by Carroll and Sutcliffe as a means of expressing an artifact's psychological consequences (*claims*) in an explicit, accumulable, and generally reusable "designer-digestible packets of HCI knowledge" [24, 29, 118, 116].

However, we have recognized the need for an evaluation tool that would allow application generic evaluation of a wide variety of notification systems. As part of the vision that we initially described in grant proposals and a pair of workshops at the UbiComp 2002 conference [33] and CHI 2003 [20], the prospect of a generic evaluation tool was perhaps met with the most encouraging positive feedback. The key benefits of an application-generic evaluation tool include the ability to easily compare and benchmark system performance, recognize progress toward reference tasks [133], and collect experience necessary for cost-benefit reengineering assessments. Although we initially had concerns about the richness of evaluation results that could be acquired through using an evaluation tool that was not tailored to specific interface features and claims, our recent study that compared specific and generic usability evaluation tools for notification systems determined that generic tools can validate or mitigate a claims analysis just as well as a specific tool [115]. Other recent and proposed efforts are moving toward evaluation tool development for specific classes of systems, such as ambient [82] and information exhibit [113] heuristics, and even heuristics for any notification system [15]. While these efforts show promise, none have established tools that can be readily applied in a usability study, allowing results to be scored and compared in a standard way.

Claim 5 – A generic usability evaluation tool for a wide variety of notification systems…

+ met with encouragement of grant reviewers and workshop attendees

+ allows recognition of progress across multiple design toward benchmarks and reference tasks

+ enables reliable cost-benefit reengineering assessments

– BUT, may result in various controversial issues → Critical parameter issues, **Section 6.2**

User's model IRCs can abstract the effects of notification design on the user's dual-task experience. Since an IRC rating only expresses supported levels for user interruption, reaction, and comprehension, and a set of interface usability results would typically include much more information about the usage experience, this implies a reduction of usability data. Like the reduction of user goals and situational variables into the design model IRC, simplifying usability data to express a user's model IRC allows focus on critical parameters that are common to any notification system. Certainly, it will be important to preserve other usability findings and empirical factors (e.g. task information and situation variables) that do not directly contribute to IRC values. Specific usability results that lead to IRC values, while perhaps not immediately implied by the parameter value, can also be stored as supporting claim tradeoffs. However, the concise user's model IRC value allows simple organization of design options that can add much efficiency, flexibility, and creativity in the development of a design.

Claim 6 – IRC user's models express dual-task usability effects of notification systems ...

+ abstracts richer data related to system usability that can be preserved in claim tradeoffs

+ channels focus to artifacts that impact goals common to all notification systems

+ informs the designer about missed or incorrectly realized design objectives, when compared to the design model IRC

− BUT, no tools available to provide user's model IRCs

IRCresults, **Section 6.3**

Using a user's model IRC as an index for interface design strategies (e.g. use of animation, color, or audio) or complete applications allows a designer to inspect design options that produce the same general effect on notification system critical parameters. Designers can then weigh these options according to platform suitability and other information expressed in associated claims. Results from studies of specific information design strategies can be reflected with multiple IRC values. For example, any of the studies presented in Chapter 3 can be reduced to IRC values that are associated with specific information design variables. A study that investigates the effects on users caused by subtle variations to information display (e.g. changing display size and speed for tickers, blast, and fade animation [85]) can be captured with user's model IRC variations so that key findings for notification systems can be quickly recognized and compared to other options by designers. However, the prerequisite for this utility is a standard method of calculating a user's model IRC from usability results with sufficient accuracy and consistency.

Demonstrating that the IRC framework can facilitate comparison of a notification system design and user's model–even the attempt to engineer a process that pushes this literal interpretation of Norman's cognitive engineering argument [100]–provides a unique and valuable HCI and usability engineering method. Being able to compare these two models will allow several design cycle efficiencies. Of primary interest in this research phase, designers will be able to explicitly realize the distance of their actual design from the intended design. This could prompt a conclusion that the design is ready for implementation (because the models match), that the design model was poorly formed (because there is a distance between the models, but the user was satisfied), or that the design must be reengineered according to certain parameters (also because of a model disagreement, but as a result of a dissatisfied user during user testing). While this method will not provide insight into all notification systems usability concerns, it will address the critical parameters of notification design–those that directly impact the attention-utility theme (discussed in Chapter 4). Although we speculate that comparison of these two models can lead to other design

Claim 7 – IRC user's models can be used as an index for notification interface design strategies …

+ enables designers to identify artifact options that produce same general effect

+ provides a concise, "digestible" summary of research findings in a comparable manner

− BUT, requires a sufficiently accurate and consistent calculation method

− BUT, implies that our understanding of IRC cannot evolve

IRCresults Studies, **Sections 6.4, 6.5**

Intended Use, **Section 6.3.2**

efficiencies, we will develop those arguments in Sections 6.2.1 – 6.2.4.

6.1.1 Problem statement and general approach

The IRC framework can allow usability measures to be described in concise terms that facilitate comparison of design effects on users and ultimately achieve realization of a design's success. The specific question addressed in this phase of the research is:

Can a user's model IRC meaningfully summarize usability test results to provide a user's impression of a notification system interface?

In order to achieve this design cycle efficiency for notification systems, we recognized that two additions to the IRC framework were necessary. The first step required creation of a reusable evaluation tool that could probe key usability concerns in early design stages for a wide variety of notification systems. Next, we needed a mechanism to convert the possible evaluation outcomes to IRC factors, providing a seamless, automatically generated user's model IRC. Since these efforts would also require testing to validate the effectiveness of the additions, the final step consisted of an experiment. The remainder of the chapter describes each of these steps in detail.

6.1.2 Expected results and successful endstate

We expected that this project would produce a generic tool that can be used to assess usability of notification systems in early design phases. Early versions of the tool need to be tested with actual evaluators and users on multiple systems. General documentation of the important considerations should be archived in as something like a checklist for the evaluation of notification system evaluation tools. Certainly, with the successful conclusion of this project, we wanted the tool to be ready to be confidently used in a notification system's usability test. Achieving this successful endstate implied obtaining a measure of the IRC's reliability in providing a comparison framework for notification system design and user's models.

6.2 Issues with Using Critical Parameters

Although we have made progress through understanding and articulating notification systems design challenges in terms of design model IRC parameters (as discussed in Chapter 5), as we continue to develop IRC assessment tools, we feel important counter-arguments must be continuously acknowledged and addressed. Our sincere hope is that the analysis and potential approaches we suggest will continue the dialog on methodological and practical aspects of using critical parameters in notification systems, and interactive systems design in general. We have intentionally developed our proposal to serve as an open, corrigible record of issues and possibilities, rather than a final solution.

To briefly review, Newman introduced the concept of *critical parameters* for HCI as a mechanism to enable meaningful modeling and execution of usability evaluations that would allow systems to become progressively better [97]. These figures of merit, when defined and adopted, would help interface designers recognize the broader intentions of the technology, shifting focus away from interface-specific details to qualities that could be directly measured, compared to benchmarks, and reengineered to better serve a user's purpose. Critical parameters have three essential characteristics: their satisfaction is critical to the success of the system, they are persistent across successive systems, and must be manipulable by designers [98]. Newman presents arguments for adapting design practice with critical parameters, which others have extended as an approach for increasing cohesion and relevance within HCI research communities [133].

Certainly, critical parameters are similar in concept to usability specifications in software interfaces, but there are key distinctions as well. Usability specifications, according to [61], are quantitative goals that a user should be able to achieve through the use of an interface. For example, the learning time for a specific task should not exceed a certain threshold, or a particular task should be able to be completed error-free by a trained user in a certain amount of time. Usability specifications are intended to focus development and ensure that objective measures are defined for summative testing. While usability specifications may be set according to performance attained in similar competitive systems, they are most meaningful when applied to a specific interface as a judgment of "goodness."

In contrast, critical parameters are intended to recenter a usability engineer's focus away from interface-specific qualities and toward usablilty goals that would be common for a large class of system. While usability specifications serve the immediate needs of the interface designer, critical parameters serve long-term research growth as an impetus for innovation [97]. This far-reaching intent of critical parameters naturally weakens the task specific utility provided by usability specifications, making the concept cumbersome for its primary users and suggesting the need for improvement.

This chapter demonstrates an extension to the concept of critical parameters, adding back some of the utility of the usability specification approach but preserving the power of generalizable user goals and performance characteristics. As the benefits of critical parameters are promising for, and perhaps most adoptable in a newly emerging design research area like notification systems, we have embraced them fully. To reflect on essential background related to both notification systems design challenges and the tensions surrounding our critical parameter-centric approach, we introduce the key issues around potential design-related capabilities that emerge with a proposal for critical parameters. Many of these issues have been introduced by anonymous reviewers, workshop attendees, and HCI students reacting to our work. We strive to present all major argument tradeoffs that have come to our attention—mitigating the downside points provides a basis for our continuing proposal and much future work.

6.2.1 Critical parameters can... create scenario families

As we seek to define and establish critical parameters for a class of systems, it is important to explore the coverage of systems for various combinations of parameter values. As different systems will be used in similar ways, it is useful to have a mechanism for capturing the similarities.

Critical parameters support the organization of systems by *scenario families*, collections of systems and the context of their use grouped by critical parameter value combinations. Including not only a description of the system but also a description of its use suggests meaningful critical parameters for a design class by shifting focus away from just the technology onto its use. This allows abstraction of the problem space and efficient focus on key design concerns.

However, the use of scenario families risks limiting novel thinking and innovation in the design of new systems, particularly those that use emerging technologies. It may be difficult to generalize lessons across platforms, information types, and other usage situation particulars. By their very nature, scenarios focus a reader on a very specific situation, and great care must be taken in constructing a scenario family to achieve appropriate coverage of the wide range of systems that should be included in it.

6.2.2 Critical parameters can... form a general design space

An important step in design and knowledge reuse is the categorization of systems in a domain. Scenario families exemplify key collections of systems, but a definitive design space should position all systems within the space, organizing all existing efforts as a body of examples. In so doing, the space allows recognition of research and innovation opportunity using common critical parameter values. While no design space can capture every possible concern that a designer or user might have, by locating all systems (and their component artifacts) within a general design space we subscribe to the belief that some knowledge is better than none at all—a developer can use the space to focus thoughts, guide decisions, and build on the work of predecessors.

However, the difficulty still arises in that we may not have a key, manageable set of critical parameters. It has proven difficult to define commonly used terms in a way agreeable to all even for a new domain like notification systems—for more mature disciplines, it may require an impetus that rarely occurs, such as a dynamic intellectual leader or a large and focused monetary commitment.

Even when a group of researchers agree on critical parameters, there is a need to be able to consistently quantify parameters on a scale. However, it may prove difficult to do so with parameters that are generally considered abstract or nonlinear, such as distraction or privacy. A tradeoff occurs when parameters must be unpacked to the point where the relationship between them is clear—terms are simplified and dependencies removed, but the important broader concept can be obfuscated.

6.2.3 Critical parameters can... express problems

Designers often face a difficult task in addressing unfamiliar problems that arise in the design process. Expressing new design problems in terms of critical parameter values allows efficient association with theories and guidelines from psychology, sociology, and human factors-information that is otherwise difficult to obtain. Designers are, in effect, using critical parameters as an index into a vast store of knowledge.

However, this process again relies on agreement with and consistency of critical parameters. In their current form, designers must know, understand, and accept the critical parameters of a field to benefit from them. Also, one can argue that this process minimizes the skills of designers, who currently access this information intuitively. For such designers, the formalisms of critical parameters threaten to stifle creativity and waste time, and are therefore viewed as unnecessary overhead.

6.2.4 Critical parameters can... support mediated evaluation

Mediated evaluation builds a store of knowledge through the design process by creating goals early on, then augmenting or modifying them through the design process to keep work focused on the needs of the user and to understand where the value of the final product resides [29]. Assessment of critical parameter values through mediated evaluation can allow systems to be compared in formative phases with other systems, benchmarks, and standards. As the development process progresses, incremental improvements through *hill climbing* [24] can address the weaknesses of the developing system with respect to the parameters identified as most important, thus lending a systematic structure for knowledge accumulation and reuse.

However, mediated evaluation based on critical parameters relies on standard, unavailable assessment and classification techniques. In addition, the processes related to mediated evaluation are not yet well understood, and the standardized assessment techniques may be limited in generality by platform and usage situation particulars, requiring significant effort in the evaluation phase. Designers want to evaluate interface features that are important to them, not ones that are important for the research community.

Table 6.1: *Proposed critical parameter components.*

	Abstract term	Concrete term(s)
General purpose	• Summarizes a user goal • General psychological/ human information processing effect • Meaningful across situations and platforms	• Measurable with an instrument • Manageable through design changes • Characterizes a specific instance in a suitable context
Necessary for…	• Defining design spaces • Requirements engineering • Reusing designs • Comparing interfaces	• Testing artifacts • Explaining effects • Preserving context

6.2.5 Residual issues

Having recognized these and other challenges in using critical parameters for design knowledge reuse, this section explores the key outstanding problems. While our approach is not intended to be a final solution, it should evolve thinking and be exemplary of what can be done in the field.

A more specifi c problem statement. Before critical parameters can be used in notification systems development to capture design knowledge from usability testing, at least two important issues must be resolved. First, it is unclear how an approach for classifying usability artifacts according to critical parameters would proceed. While it may be possible to put forth general artifact characteristics that merit certain ratings and assist classification efforts (i.e., "fast tickering rates have high interruption," or "audio cues provide low comprehension"), this approach would be mired in subjectivity or require an unwieldy set of platform-specific guidelines. Furthermore, it would close dialog that would be useful for conceptual evolution of the critical parameters, their definitions and scales, and measurement techniques. Therefore, a second important issue is determining how classification approaches can encourage critical parameter conceptual evolution.

Conjecture and argument structure. Clearly, if it were possible to express notification system design challenges in terms that anyone could understand (akin to usability specifications)—and readily compare—we would gain many advantages. In order to achieve this, we propose that a critical parameter should have two parts (shown in Table 6.1).

- A sufficiently *abstract term* to allow meaningful generalization and express user goals and situational expectations, and
- *Concrete term(s)* for measurable and manageable psychological effects that can be directly observed or estimated for a given artifact.

Elaborating our previous idea of critical parameters in equation form demonstrates this conjecture, and provides resolution to many of the residual issues inherent in our approach. This general approach evolved naturally from our development and tuning efforts with the IRCspec algorithm (see section 5.3.2).

Argument in support of this proposal proceeds in the following sequence. First, we show how equations unpack the current critical parameters and provide both abstract and concrete facilities for characterizing notification systems usability concerns. Component variables assist in defining abstract parameters, providing a means for generality and reuse, as well as measurability and manageability. Second, we illustrate how critical parameter equations provide a point of convergence for a variety of usability evaluation methods and assessment instruments. We demonstrate two possible methods (analytical and empirical through controlled lab testing), and provide a case study to detail evaluation results using the analytical instrument on three different notification system interfaces. Results suggest the utility of this approach based on critical parameters, and indicate that we are able to make progress toward using the approach with HCI education efforts. We speculate about other broad implications.

This argument addresses a few of the key concerns raised, but leaves other concerns for future work. In particular, future efforts must address generalizing claims to extend proposals by Carroll and Sutcliffe [24, 29, 118, 116]. Focusing initial efforts toward structuring a design process for the benefit of HCI education diverts immediate need to address points related to designer overhead, but it is our hope that features built into an integrated development environment emerging from ongoing work will mitigate these arguments. Only time, broader dialog, and additional experience will increase or decrease our confidence in critical parameter selection.

6.3 A Generic Evaluation Tool for User's Model IRC Ratings

Throughout our work, we have proposed three critical parameters to capture user notification goals related to interruption, reaction, and comprehension (IRC) [84, 87, 89]. As the notification systems design space illustrates, systems can be thought of as having targeted (*design model*, as in [100]) and actual (*user's model*) values for each parameter. For example, a stock ticker notification system may be designed to target low interruption, low reaction, and high comprehension (the ambient class)—but actual system usage may display a complete inversion of these parameters (the alarm class) (see Chapter 4 for a full discussion of the IRC critical parameters). Understanding targeted goals and user performance characteristics in terms that are comparable to each other and other systems provides opportunity for many benefits, but abstract parameters must be associated with concrete terms that can be assessed in usability evaluations.

6.3.1 Relating abstract and concrete terms with equations

Three equations are introduced for notification systems interface evaluation, allowing conversion of measurable, manageable concrete variables (summarized in Table B.2) to the abstract parameters that relate to general user goals and psychological effects. This effort extended naturally from the development of the IRCspec questions and algorithm—as we formalized our thinking about the

$$\mathbf{I} = 1 - s^{3 \cdot COI}$$

where s = sustainment
COI = cost of interruption

Figure 6.1: Equation for the Interruption (I) critical parameter.

underlying concepts, the variables, variable values, and variable relationships emerged. Figure B.1 revisits the IRCspec questions that are used in the calculation of a design model IRC, associating concrete terms and values with the answer choices.

The equations are not necessarily intended to be a robust, integer-based system. Instead, the equations are intended as a conceptual metaphor, loosely organized as a categorical, interval scale approximation. When considering the validity of the equations, one should think of them as numeric representations of *low, somewhat low, moderate, somewhat high,* and *high* parameter categories. The equations are thought to assist in obtaining more consistent selection of these concrete categories while assigning abstract user's model parameter values. Numeric representations are useful in facilitating search/indexing operations. The case studies in next two sections present related test results.

Interruption. The first critical parameter we have identified for notification systems design is *interruption*. There have certainly been many important branches of work in cognitive and experimental psychology to understand the facets of interruption, and recent efforts within the HCI research community have helped deliver findings to system designers and evaluators [7, 41, 65, 92]. Seeking to improve this transfer of research findings, we offer a simplified model of interruption suitable for design and evaluation of notification systems (see Figure 6.1).

In this conception, interruption (I) can be described as the effect of reallocating attention from the primary task to the notification. "I" describes both the appropriateness of an interruption, as well as the actual interruptive effect of the notification artifact (distraction to the primary task). Therefore, "low I" can describe either an artifact that supports attention grading/parallel processing during the performance of an urgent primary task (high sustainment, regardless of COI) or any quality of multitasking performance in a non urgent situation (low COI, regardless of sustainment).

Appropriateness of an interruption is represented by COI (cost of interruption), characterizing the user's willingness to accept an interruption, and thus the urgency of the primary task can be inferred. As established by Horvitz's Interruption Workbench [65], COI describes a total task situation in terms of how much a given user would typically pay in dollars not to be interrupted. The Interruption Workbench records a variety of situation characteristics, such as the specific primary task application, level of ambient noise, recent keystroke and mouse activity, etc) over an extended period of normal user activity. The tool segments the observations into periods in which the task variable combinations are consistent. Users rate each segment, assigning the dollar value they would pay to avoid interruption, allowing Bayesian inference networks to aggregate samples and determine probability distributions for various costs of interruption levels. Alternately, this value can be estimated based on existing empirically determined examples.

Regardless of the assessment method, the cost of interruption analysis should be a function of the cost-benefit relationship inherent in the dual-task situation. This implies a consideration of both the strength of the pull factor of the notification, or how much a user is likely to need or want the information contained in the notification (benefits), as well as the costs to the primary task. Costs are elevated when the primary task is more important or urgent (thus, consequences of performance degradation are critical), when it is not data-driven by the notification system, and when working memory at the time of a notification is expected to hold information that is the result of a series of complex information processing cycles.

Therefore, as one calculation method, we can construct several scenarios of typical primary task interaction, as well as several scenarios that describe introduction of a notification and brief interaction with a notification system. Each notification scenario can be characterized by the basic benefits, and then paired with all possible primary task scenarios. Keeping in mind the benefits, the evaluator can then assign the costs of interruption. COI values for scenarios can be averaged and weighted appropriately, or kept separate to show the range of possible I-values.

Actual interruptive effect (s) can be gauged by primary task sustainment—a metric used to quantify the change in the primary task performance from solo-task to dual-task performance. Calculation of primary task sustainment has been demonstrated for notification interfaces [120] and broader psychology efforts [135]. For example, if the primary task is editing a document, an evaluator can observe the solo-task performance characteristics related to a user's editing speed (how much of a document was a user able to edit in a given period of time?), accuracy (what is the level of revision technical correctness?), and thoroughness (what portion of known errors are identified and revised?). The same performance characteristics can be observed in a dual-task situation (in which the notification system is also being monitored). An evaluator will have to use their judgment to select performance measurements that characterize usage situation at appropriate granularity—second-to-second observations may be best for some situations while other situations should be modeled with observations taken at less frequent intervals. Dividing the solo-task measures by the dual-task measures provides sustainment scores that can be averaged with weights that are appropriate to the situation context. This score provides an indication of the level of primary task performance that is typically sustained while a user is simultaneously monitoring or interacting with the notification system.

The equation we present is modeled with an exponential COI to reinforce the importance of this factor, but tripled to ensure a fairly wide range of I-values for a given COI and to produce a moderately high I-value (0.65) when both s and COI equal 0.5. By comparing the design model I-value (assessed by reviewing requirement specifications) with the user's model I-value (assessed by analyzing the effects that result from the actual system), a designer can determine the extent to which the interruption caused by the notification is appropriate for the given usage situation.

Reaction. The second abstract critical parameter term for notification systems, *reaction*, describes a user goal that can be generalized as an immediate response to a new notification.

Table 6.2: *Concrete terms used in the I equation, with usability evaluation assessment techniques for each.*

Concrete Term		**Assessment Technique**	
Symbol	**Description**	**Analytical/Subjective**	**Empirical/Objective**
COI	cost of interruption	*Given the nature and importance of the user's primary task at the receipt of the notification, how* ***costly*** *would an interruption be?* {extremely = 1; very = .75; moderately = .5; not very = .25; not at all = 0}	Interruption Workbench output; P(High) is weighed at 1, P(Med) = .5, P(Low)=0
s	primary task sustainment	*Compared to the primary task performance before the notification delivery, how much does the primary task performance* ***reduce*** *when the notification is present?* {not at all = 1; less than half = .75; about half = .5; more than half = .25; completely stops = 0}	Ptask performance while multitasking divided by ptask performance as a solo-task

$$\mathbf{R} = \frac{(t \cdot h)^{\frac{1}{3 \cdot COI}}}{2} + \frac{h(0.5 + COI)}{3}$$

where $\boldsymbol{t}$ = relative response time
$\boldsymbol{h}$ = hit rate

Figure 6.2: Equation for the Reaction (R) critical parameter.

The reaction (R) equation (see Figure 6.2) consists of two parts, each worth up to an R-value component of 0.5. The first term takes two reaction performance metrics—hit rate (h) and relative response time (t)—and lowers the average according to strength of COI. *Hit rate* refers to the concept from signal detection theory [53] where a user correctly detects and responds to a signal (a notification). Relative response time is a ratio between actual and expected response times. Certainly, expected response times may be dependant on usage context and information characteristics, and they should be estimated or obtained in requirements gathering. This estimation can be based on GOMS analysis or appraising benchmarks achieved in existing systems that are similar. Even a rough approximation of expected response time can be valuable for comparing the actual response times. To calculate hit rate, an evaluator divides the total number of hits by the total number of signals. A similar metric, false alarms (or false positives), which refers to the comparison between hits and total responses, is used in the comprehension (C) equation.

The second term of the R equation can add up to half the hit rate to the R-value, depending on the strength of COI. Moderate reaction (R=0.5) is scored when two-thirds of the hit rate and reaction time is achieved with a COI of 0.5. Moderate or high R-values are always obtained when one of the variables is near maximum and the others are at least moderate.

The equation is also designed so that no more than R=0.5 can be achieved if one of the three variables equals zero. In order to understand this rationale, one must consider that R is a characterization of an artifact's effectiveness for supporting reaction in a dual-task situation. That is, if the notification system is not attempting to resolve a situation constrained by the tradeoff of limited attention for gain in utility (the attention-utility theme [89]), in which there would generally be at least a moderate value for COI, then the appeal of the artifact for facilitating notification reaction in a dual-task situation is inherently limited and therefore penalized. Both aspects of the reaction performance are also critical-a near-perfect hit rate would not be looked at as effective reaction if the response time were significantly slower than specification. Likewise, an acceptable response time has limited worth in the case that most signals delivered by the notification system are missed. Another feature of the equation is the prominence of the hit rate. Factoring this variable directly into both terms allows quick growth of R-values as hit rate increases, especially when COI is greater than 0.5. This adds a strong characteristic to R of being a measure of response selection probability.

Comprehension. Our abstract parameter of comprehension is based on the concept of situation awareness, in which a user accumulates Perception (of the elements in the system), Comprehension (of the current situation), and then Projection (of future status). Each level is dependent on achieving some part of the preceding level, and represents a progressively higher state of situated awareness [46]. Thinking of notification comprehension as situation awareness brings our efforts in characterizing notification systems in line with a wealth of research in the human factors field, and reinforces our argument that each parameter is a separable dimension. For instance, studies have shown that we can recognize the characteristics of awareness independent of the processes required to maintain it (working and long term memory or attentional state) [2] or the response selections that result from it [133]. Thus, the comprehension critical parameter describes longer-term

Table 6.3: *Concrete terms used in the R equation, with usability evaluation assessment techniques for each.*

Concrete Term		Assessment Technique	
Symbol	**Description**	**Analytical/Subjective**	**Empirical/Objective**
h	hit rate	*How often will users actually notice important changes in the notification, as opposed to not noticing them?* {always = 1; more than half = .75; about half = .5; less than half = .25; never = .0001}	As in Signal Detection Theory, P(H) divided by total signals
t	response time	*In cases where a notification suggests an action for a user to take, how does the user's response time compare to the reasonably desired response time?* {better or as good as expected = 1; slightly slower = .75; about twice as slow as expected = .5; much slower = .25; extremely slow or action never taken; no action ever required = 0}	Determine *actual response time* (*a*) as the difference between signal presentation and signal response; *expected response time* (*e*) provided in system specification; *t* = *e* / *a*, when *a* > *e* (otherwise *t* = 1)

$$\mathbf{C} = f + \frac{(1-f)(p+2c-cp)}{3}$$

where $\boldsymbol{p}$ = perception rate
$\boldsymbol{c}$ = base comprehension
$\boldsymbol{f}$ = projection (future)

Figure 6.3: Equation for the Comprehension (C) critical parameter.

(not immediate) knowledge gain.

An important aspect of our decision to base the comprehension equation on situation awareness is the importance of having comprehension measures that are closely tied to the usage situation. Comprehension describes the extent to which a notification system imparts (and is expected to impact, in the case of the design model) understanding of new information. Such understanding can be determined based on the actions taken (or not taken) by a user, cued recollection of states and trends, and accuracy of inferences based on the notification information. Certainly, an evaluator must make these judgments based on information that is inherently part of the task context; however, the C equation provides a structure to map this specific, task-related assessment to generic terms that can be more broadly compared.

$$\mathbf{C} = \frac{p+(1-p)(c+f(1-c))}{3} + \frac{c+f(1-c)}{3} + \frac{f}{3}$$

Figure 6.4: Unsimplified version of the C equation

The simplified equation that appears in Figure 6.3 is difficult to explain, so we show the unsimplified version in Figure 6.4. This equation consists of three terms-one for each level of situation awareness (perception, base comprehension, and projection). As each level is maximized, the equation ensures that C=0.33, 0.67, and 1 respectively. If a given level is not maximized, achievements in the higher levels provide "credit" toward the C-value.

As mentioned earlier, *perception* refers to the notification system's support for invoking signal hits rather than false alarms (when a user responds to something that was not a signal). *Base comprehension* requires interpreting and remembering characteristics of the current state depicted by notification information. Defining general levels of base comprehension requires an evaluator to reflect on system requirements and identify quantities and qualities of recall that would typical. As performance benchmarks are established, this portion of the test design can be more consistent, but the definition of "high" or "low" base comprehension should be influenced by the situation context more than anything else. *Projection*, the highest level of comprehension, refers to a user's ability to make reasonable inferences about future states of information that might appear as notifications. That is, based on their understanding of the current state, recent history, and trends observed over time, a user might be able to predict how and when the information of interest will change. Again, testing users on ability make projections requires careful consideration of the system requirements and situation context—it may be entirely possible that a system is not intended support such goals, and therefore impractical design test items that would generate a full projection rate.

Still under review is the issue of whether COI should be an additional factor in the C equation. Some justification for this is present in Endsley's argument that temporal dynamics play an important part in assessing the comprehension and projection levels. Specifically, she mentions that part of projection requires an understanding of the rate at which information is changing. However, by articulating the concrete terms we rely on to form our abstract notion of notification systems comprehension, we open this issue and others for debate within the research community.

6.3.2 Intended use and evolution

As stated previously, we present these equations as a conceptual metaphor to connect concrete critical parameter terms with abstract terms that can be generalized to understand design spaces, facilitate requirements engineering, support design knowledge reuse, and compare interfaces within a common design domain. Each variable on the right side of an equation is a concrete term that can be measured in requirements gathering and usability testing with a wide variety of methods, as we demonstrate in the next two sections.

Abstract and concrete terms for critical parameters like these can be introduced for any other class of interactive system to describe user goals and psychological effects of the interface. We hope that our community of researchers will work to evolve these conceptions, adapt them to their own needs, and ultimately improve consensus. Thinking of these terms as "slots" to guide discussion within the research community, we see an important opportunity for mechanisms that elaborate and validate relationships between variables, as well as research that demonstrates extensible, context-specific assessment methods for obtaining concrete variable values.

Table 6.4: *Concrete terms used in the C equation, with usability evaluation assessment techniques for each.*

Concrete Term		Assessment Technique	
Symbol	**Description**	**Analytical/Subjective**	**Empirical/Objective**
p	perception rate	*When considering the total number of times a user interacts with the notification system, what is the* ***ratio*** *of the interactions in response to an important notification vs. total interactions (including those when no actual notification was being delivered, i.e., user checking on their own or thinking there was a notification)?* {1 to 1 = 1; 2 to 3= .75; 1 to 2 = .5; 1 to 4 = .25; more than 1 to 4 = 0}	As in Signal Detection Theory, P(H) divided by total responses
c	base comprehension	*How much of the notification content will the user want to remember and be able to remember several minutes after the notification is delivered?* {all content = 1; more than half = .75; about half = .5; less than half = .25; none at all = 0}	Quiz user on a sample of notification content questions to assess correct interpretation, relationship to goals, and storage in long term memory. Use % correct.
f	projection	*Based on the notification content, how successful will the user be in making projections or predictions about future trends or the long-term state of the system being monitored?* {extremely successful = 1; very successful = .75; somewhat successful = .5; not very successful = .25; not a goal for this system = 0}	Quiz user based on a sample of interpretations that can be projected to predict future states or notification patterns. Use % correct.

6.3.3 Two methods for obtaining IRC variables in usability evaluations

If usability evaluation activities were focused on assessing concrete critical parameter terms to yield abstract characterizations, equations like the ones we introduced would provide a point of convergence for a variety of usability evaluation methods and assessment instruments. We certainly feel that a variety of methods and instruments (along with an evaluator's indispensable expert judgment) will always be necessary for the wide ranging and continuously evolving facets typical to usage settings and interface platforms. To clarify, we discuss two possible methods (analytical and empirical through controlled lab testing) for obtaining the concrete terms in our equations.

An empirical method. Since the equations are intended to characterize the user's model of the notification system interface, many would argue that data obtained from a user's actual usage experience is of primary value. System event logging, user observation, and user surveys can be tailored to collect data for each of the seven metrics. As discussed earlier, COI can either be collected by a tool like Horvitz's Interruption Workbench [65] or a survey method with less overhead. Notes for empirically obtaining each of the variables are summarized in Table B.2. In ongoing work, we are experimenting with an automated notification systems testing platform that allows user event logging of critical actions, such as performance on a primary task with and without the notification system, response accuracy and timeliness to notification signals, and comprehension of important notification information after an extended period of time. The second case study in section 6.5 describes some of our pilot testing results with this system. Notification systems researchers have used similar testing platforms [7, 13, 41, 84, 92], and we are encouraged that data necessary for obtaining the critical parameter terms is often collected by most researchers, implying that existing experimental platforms could be easily modified.

From our reflection on empirical test instruments that help obtain the concrete parameter terms, we also note several points of caution. Since the test protocol relies on a definition of total number of signals present, evaluators should ensure users are only expected to respond to a realistic number of important notifications. This consideration may become important because analysis of signal detection performance may require that system interfaces are tested and compared based on a known, cached set of notification data to allow signal introduction times to be recognized, observed, and automatically processed by a testing platform. Alternately, user performance with actual, real-time data can be measured using screen recording or videotaping of a usability test session.

A final aspect to note about empirically assessing the concrete variables relates to the comprehension and projection terms in the C equation. We suggest data for these variables be collected in free-form or cued post-test surveys that probe recollection of key events, information states, and notification patterns. However, an important consideration in designing the test flow is ensuring that the length of the rounds (after which comprehension questions are asked) does not impose unrealistic memory reconstruction expectations for information characterizing the current state depicted by the notification system. Alternately (and less desirably in a dual-task test situation), popup windows or brief halts of the interface usage experience by the evaluator can allow comprehension-related questions to be asked throughout the test. A response mechanism that discourages participant guessing or uncertainty, such as open-ended questions or fill-ins, is particularly critical for obtaining these terms.

An analytical method. While empirical data may be preferable for characterizing the user's model of an interface design, empirical testing often comes at a much higher cost. To support user lab testing or field studies, systems must be fairly robust and further along in the design cycle (implying higher cost for large changes and sometimes preventing formative testing). Other drawbacks include overhead involved with system logging or session observation and recording, participant recruitment, lab access, and other factors. For these reasons, and to facilitate formative and mediated usability evaluation, we were eager to develop an analytical testing method that could yield terms for the concrete critical parameter values.

As Table B.2 shows, we were able to formulate a survey question and appropriate set of responses to analytically assess each concrete variable present in the equations. Just as with other analytical evaluation methods, we do not intend that a survey composed of these questions be used to collect opinions of general users. Rather, this instrument should be used by interface experts or at least experienced notification systems designers familiar with applicable challenges. While response selections provide feedback in the form of critical parameter values, perhaps of equal or greater value are the specific comments and rationale behind each rating, which can be expressed as claim upsides and downsides. We envision this analytical instrument to be used in a moderated evaluator discussion session that may or may not include the system designer, although each evaluator would provide individual assessments of each question.

The case study presented in the next section was conducted with the analytical instrument. In the case discussion, we provide additional details about the method execution and results analysis, as well as observations related to variations in session moderation techniques. We are generally pleased with the evaluation outcomes provided by this method, and recommend it as a tool for evaluating notification systems, allowing data necessary for the equations to be obtained.

6.4 Study 1: Analytically Evaluating Redesigns of The Scope

We challenged a group of novice designers to improve upon a notification system interface developed by a Microsoft Research group [127]. The Scope (shown in Figure 6.5) is a small display that resides in the corner of a user's desktop, depicting new and existing notifications in quadrants for email, calendar, task, and alert items. As a circular-shaped interface, the Scope leverages a radar metaphor to convey relative item urgency. In their research, the original design group noted several usability concerns, so we instructed the new teams (15 total) to improve upon these and other issues they discovered through their own requirements gathering efforts. The three-month redesign effort was controlled through class specifications that required a mediated approach to advancing design rationale and making interface improvements.

Motivated by their requirements gathering results rather than any instructions, several of the teams came up with very different display and interaction strategies for the Scope redesign, abandoning the radar metaphor. We wanted to compare these redesign options according to impact on notification critical parameters, and understand where each system within our design space. Other objectives of our study were to assess the difference between design model and user's model critical parameters for each system. We hoped that quantifying the conceptual models would help to expose interface features that should be redesigned in subsequent versions, suggest additional requirements gathering steps needed, as well as classify design artifacts for reuse. Note that these objectives are functions of the abstract critical parameter terms, as summarized in Table 6.1. We hypothesized that our analytical testing tool would be able to test all system designs so they could be meaningfully compared—highlighting differences between systems and between product and designer's intention (regardless of differences in display and interaction strategies).

6.4.1 Testing and analysis procedure

The first step in our testing procedure was to collect design model intentions in the form of targeted IRC values from each system's design team. We tried to get as many of designers from all 15 teams to specify their design model. Design model specification was accomplished with the IRCspec tool that had already been validated to produce accurate and consistent design model IRC values [32] (see Chapter 5). After the designers answer general questions about the dual-task situation requirements assumed for the design constraints, the tool calculates the targeted IRC values. The data collected from this portion of the study can be found in Figure B.6. A total of 38 designers completed the design model assessment, with at least one designer from each team and 13 teams with more than one designer using the IRCspec tool.

Analysis of design models. Encouraged by such a strong response, we conducted an opportunistic analysis on agreement tendencies between an individual and their team members, and the group of designers overall. Our hypothesis was that a designer using the IRCspec tool would would generate a design model that was more consistent with their own group than the average IRC values generated by all designers in the test. This hypothesis was somewhat risky, since all 13 designs (corresponding to the teams that had more than one designer submit IRCspec results) were adaptations from a single system that had a coherent IRC-related design model. However, we expected that teams would have a different understanding of the requirements relating to user IRC goals. If the output generated by the IRCspec tool could detect these differences, this would certainly strengthen our general argument that IRC values express critical differences between system design intentions.

A second analytical approach probed the interrater reliability of the scores within the design teams. Following a standard approach for constructing a data set to test interrater rating correlations, we paired parameter ratings from participants with odd and even participant IDs that were within the same design team. Unfortunately, if teams had three responses, the last one had to be dropped. Kendall's coefficient showed a non significant level of overall interrater reliability ($\tau = .18$, $p = .08$). Closer inspection of the average mean differences for each team showed that Team 3 was a legitimate outlier (their average mean difference exceeded nearly two standard deviations of the average mean difference of teams). Therefore, we dropped Team 3's data, reran the Kendall coefficient test, and obtained a significant result ($\tau = .23$, $p = .04$). The two data sets supporting this analysis can be found in Figure B.7.

Based on both tests, we are encouraged by the expressiveness of IRCspec and the IRC framework in general. Even though the designers were not prompted to focus their requirements analysis and early design rationale development on IRC-related issues, the IRC still serves as a mechanism that could differentiate design models of a very similar set of notification systems.

Three interfaces for further analysis. To continue toward the main goal of the study and test the effectiveness of the analytic version of the IRCresults equations, we selected three interface redesigns that exhibited strong differences from the original Scope concept (shown in Figure 6.5)[1]. Although implementations were only in early, unpolished prototype form, we felt that each represented distinct notification strategies that would occupy different portions of the IRC design space. Like many desktop notification systems and the Scope, the prototypes sought to convert a small portion of screenspace into a glanceable information center for notification awareness. Tooltips often provide brief summaries of notification content, with further details accessible through a mouse click. *Prototype A* was inspired by a bulletin board, introducing notifications as small notes that appear in rows according to category. *Prototype B* is a vertical bar for the side of a desktop that embodies a waterfall metaphor—notification icons fall slowly down the interface as they near their due date and unscheduled items are pooled at the top. *Prototype C* represents an iconic task list divided into several categories by notification type, which users can reorder and code by urgency. If our usability evaluation goals were met, we would help designers realize inaccurate information and interaction design assumptions and quantify the different psychological effects each option would have on users.

Evaluating the three interfaces. The second step involved presenting the interface prototypes for analytical evaluation. We recruited 34 experienced notification systems designers from other groups in the class to serve as evaluators. Between three and six evaluators were organized into sessions in which one interface was analyzed with the analytical instrument described earlier and presented as a form (see Figure B.5). Although each evaluator provided individual ratings and feedback, sessions were moderated to prompt interactive discussion among evaluators about design decisions (the script used for moderating the sessions can be found in Figure B.4). This technique was used to ensure that evaluators were engaged in the process and thoroughly informed about the interface features. All prototypes were sufficiently interactive to demonstrate intended behavior. One session was conducted with a system designer as an assistant moderator-the designer explained intentions and answered evaluator questions related to specific features[2]. However, the evaluator

[1]The interface prototypes in our case study were designed by Josh Adell, John Archie, Edwin Bachetti, Niteesh Bharara, Andrew Jackson, Joey Jezioro, Abijeet Jhala, Aaron Kaluszka, Tim Fuller, Theresa Klunk, Jed Lake, and Vinay Lakahani in CS 3724.

[2]Special thanks to Edwin Bachetti for assisting the usability evaluation.

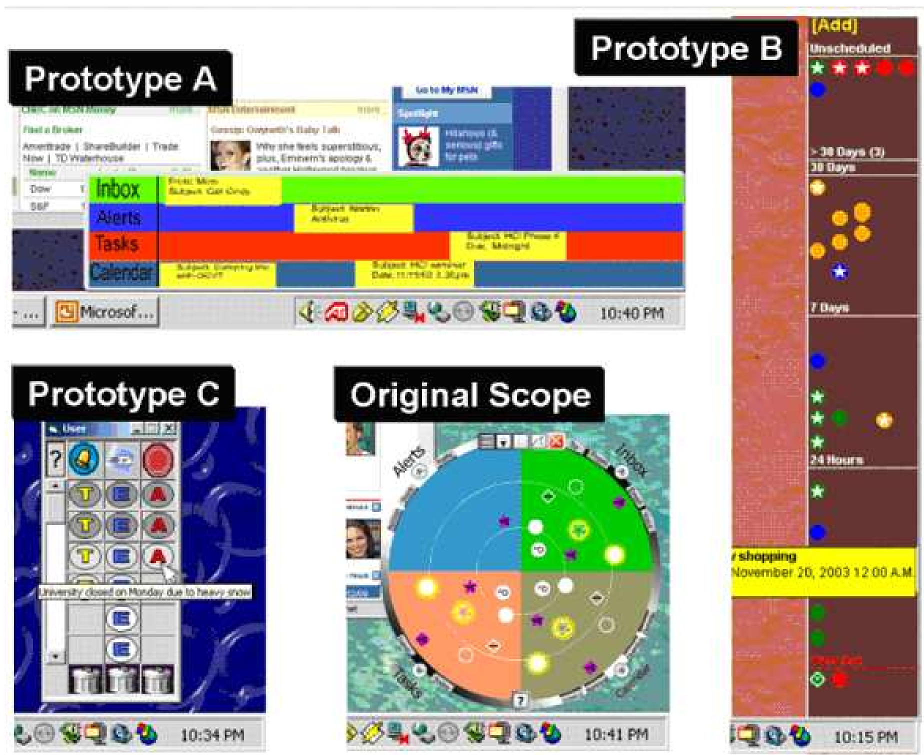


Figure 6.5: Notification system interfaces studied in the case study usability evaluation. The three prototypes are redesigns of the original Scope interface.

results obtained from this session were no more or less consistent with each other than in all other sessions, implying a negligible effect. All sessions lasted 20-35 minutes. Prototypes B and C were each analyzed by 11 evaluators, and 12 evaluators analyzed Prototype A.

Analysis of the IRCresults data. The third step was the data analysis. Evaluator responses to the multiple choice questions were entered into a tool that associated responses with values for the concrete equation terms (included in Table B.2), executed the three equations from Section 6.3.1, and returned the user's model IRC values for each evaluator. IRC values for each system were averaged by parameter, and individual response differences from the system average were compared to screen outliers. Two of the Prototype A evaluators and one of the Prototype B evaluators exceeded the threshold ($\sigma = 1.5$), so their IRC values were removed from further analysis.

Next, we checked each system's collection of IRC values to determine whether all evaluators should be grouped together when making inferences, or whether clusters of evaluators should be established. Just as differences of opinion might naturally divide a group of evaluators on key issues, we had no reason to expect that a large group of evaluators would not bifurcate just because they were using an IRCresults tool. However, we did expect to see a clear consensus (indicated by sufficient interrater reliability and made evident through more elaborate evaluator comments) that could be articulated by the critical parameters within large sub-groups.

To guide this process, we looked for the same expected rating consistency that is reliably achieved in the design model IRC assessment tool, ± 0.15 per parameter and a significant level of interrater reliability according to Kendall's coefficient τ. After removing two outliers, differences between Prototype B evaluators for all three parameters were smaller than this threshold (I_{diff} = 0.02, R_{diff} = 0.11, C_{diff} = 0.13), so the remaining evaluator results were averaged together for inferences about user's model critical parameter values (see Figure B.8). However, both Prototype A and C had one parameter each that exhibited higher average difference between evaluators. While evaluators of Prototype A were consistent about interruption and reaction ratings (I_{diff} = 0.10, R_{diff} = 0.13), they differed on comprehension ratings (C_{diff} = 0.22). Therefore, evaluator responses were clustered into two groups: those that assessed high and low levels of comprehension (see Figure B.9). Although this resulted in two sub-groups, we were satisfied that each sub-group expressed a clear and distinct interpretation of the notification system's usability. Prototype C evaluators differed on opinions about interruption (I_{diff} = 0.25), and the same clustering approach was used to establish two reasonably coherent sub-groups (see Figure B.10). With only one exception (out of 15 cases), average differences between evaluators in new clusters for all three parameters fell within threshold consistency.

Finally, we wanted to determine whether the analysis instrument provided significantly more consistent IRC ratings with evaluators assessing the same system, when compared to all evaluators regardless of system. To determine this, we pooled each evaluator's parameter differences from their system's I, R, and C averages and compared that to each evaluator's differences from the overall I, R, and C averages established by all 31 evaluations (a benchmark that would be representative of general redesign approaches to the Scope). To compare these scores (based on ordinal observations), we used the non-parametric Wilcoxon Rank-Sum Test. Our hypothesis (H_1) was:

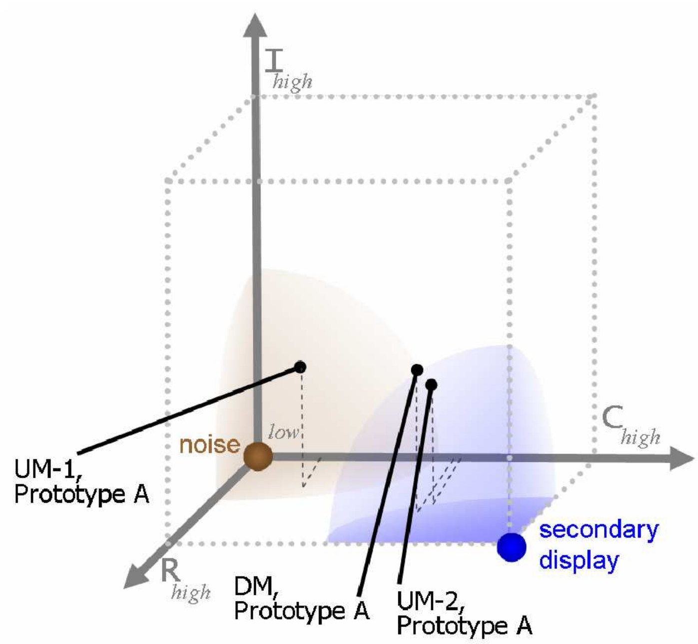


Figure 6.6: Design model (DM) and user's model (UM) assessments for Prototype A, evaluated in the case study.

The distribution of dispersion for the evaluator differences with the group as a whole would be shifted to the left (or upper-tail), indicating stronger association with team scores.

H_1 was supported by a one-tailed test on the rank-ordered data, $p = 2pr(W_{MA} \geq 10709) < .001$.

6.4.2 Study results and implications

Confident that our instrument is sensitive enough to produce evaluator results expressing system nuances, we used IRC averages (depicted in Figure **6.6-6.8**) to make inferences about usability issues and possible redesign directions.

Having collected consistent design model IRCs from the system designers, we recognize that Prototype A was intended to support moderately low interruption (I = .43, on a scale of 0 to 1), high reaction (R = .72), and moderate comprehension (C = .63), which would be a secondary display notification system with lower than usual goals relating to long-term recall of information. Unfortunately, while evaluators generally agreed that the interface support an appropriate level of interruption (I = .30 to .35), they differed in opinions related to the other goals. One cluster of evaluators (labelled "UM-1" in Figure 6.6) thought that reaction would be too low (R= .33) and comprehension would be very low (C = .17). However, the user's model ratings by second cluster agreed much more closely with the design model: I = .35, R = .54, C = .62, implying that the design may meet intentions for some users.

Mitigating the concerns expressed by evaluators in the first cluster would be an important next step for these designers. Background and demographic differences could be studied further to identify distinctions between evaluator groups. Stated comprehension concerns could also be immediately addressed with more sophisticated visualization techniques. For example, one concern involved missing new notifications entirely due to clutter, overlap, and poor scalability—a problem that might be solved with a fisheye technique. Another issue raised was a user's inability to ascertain relative urgency of notifications—a feature apparent in the original Scope that enhances reaction.

The design model IRC for Prototype B was only provided by a single designer, but reflected moderate values for interruption and comprehension (I = .37, C = .52) and a moderately high reaction value (R = .72). Unfortunately, this designer mentioned that they had played a very passive role on the design team. To gain more information about the team's design model, we conducted interviews with the two primary designers, which revealed strongly opposed views for the goals of the system. One designer thought a tool that supported very high comprehension and long-term planning would be best, while the other wanted an alarm-like system that would be used to process urgent notifications and forget about long-term action items (see Figure 6.7). While each designer thought they had compromised their goals somewhat, the first designer's model carried through to interface implementation and the user's model IRC. Evaluators consistently rated this system to be an ambient system, with moderately low interruption and reaction (I = .26, R = .27) and moderately high comprehension (C = .66). As expected, both designers were not satisfied with the evaluation result. In this case, the critical parameter models reveal a need for re-negotiation of the requirement assumptions for the basic user goals. This process can be assisted by discussing specific points on the design model survey. However, the system as it is provides a strong artifact example of an ambient user's model.

The design model for the final interface consistently targeted moderate interruption and comprehension (I = .62, C = .52) and high reaction (R = .81). According to both clusters of evaluators, the designers missed their intention. Both clusters agreed that reaction would be moderately low (R = .24 and .21), a major difference from the design model that would be essential to correct. Evaluators were concerned that new notifications would be detected too slowly, since user memory overhead would be too high without any glanceable notification context and the interface's scrolling mechanism would be problematic. One cluster saw these problems as a basis for moderately high interruption (I = .79), while the other cluster felt the interface would simply be ignored and introduce interruption less than intended (I = .29). Both clusters thought reasonably moderate comprehension gains would be supported by this interface (C = .57 and .45), (see Figure 6.7). Faced with these large disparities, the design team may be wise to consider an alternate approach.

6.4.3 Broader implications: Comparison and reuse

While the IRC parameters were useful in assessing each design individually, the broader benefits of using critical parameters are recognized in activities such as system comparison and design knowledge reuse. For example, if we are looking for a more ambient redesign of the Scope,

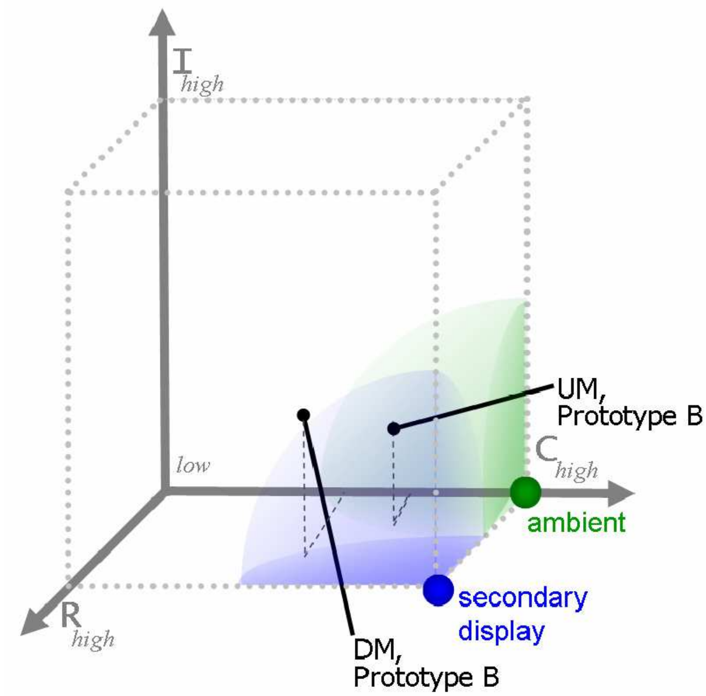


Figure 6.7: Design model (DM) and user's model (UM) assessments for Prototype B, evaluated in the case study.

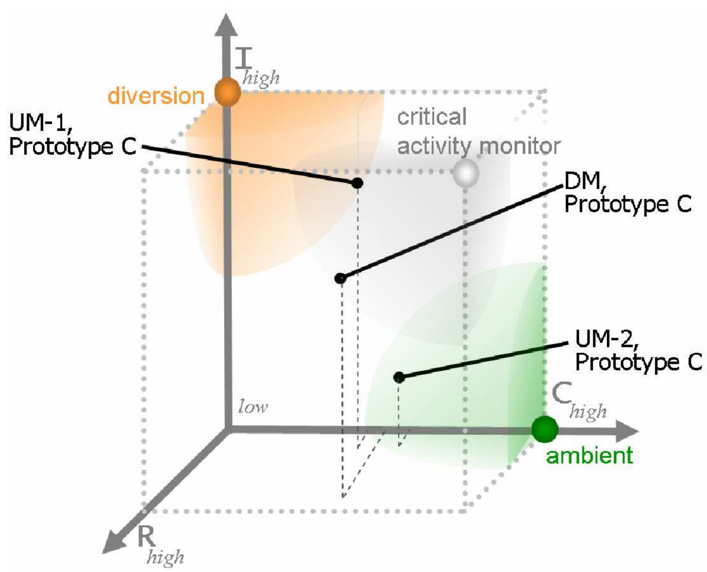


Figure 6.8: Design model (DM) and user's model (UM) assessments for Prototype C, evaluated in the case study.

Prototype B would be the best starting point. However, techniques used in Prototype A may offer relevant inspiration, and it may be wise to conduct an evaluation on the Scope to see whether real critical parameter improvements are even being proposed. As information and interaction design changes are made to any system, a series of IRC evaluations can show progress between versions, as well as specific effects of feature-level artifacts. These psychological effects can be recorded as claims [29], indexed by IRC values [101], and archived in a library for design knowledge reuse [116]. For designers that are faced with brainstorming notification options that match a particular design model (perhaps like the designers of Prototype C), such a library may be an indispensable resource.

6.5 Study 2: Empirically Evaluating New Expedia Notification Interfaces

As part of a project to encourage undergraduate students at Virginia Tech to learn the C# programming language, we created a contest that required the design of a specific notification system. The notification systems were to help users monitor and maintain awareness of airplane ticket prices for two specific destinations from four departure points, as currently listed on the Expedia.com[3] website. Designers were provided general design guidelines, such as keeping the desktop display small and suitable for persistence in the corner of a screen, minimizing user annoyance or distraction to their ongoing work, and providing an intuitive interface for typical college-aged users. The full specification for the Expedia notification systems can be found in Figure B.11.

6.5.1 Notification data and interfaces

The interfaces were required to be interactive for a 10 minute period of time and operate from a given data set (see Figure B.12). This dataset simply consisted of the cheapest ticket price currently available for each of the eight flights and the times at which each price changes. In order for us to achieve other project goals, designers were constrained in their choice of programming language to C# and were required to using at least two specific programming constructs introduced by the language paradigm. However, they were given more than 6 weeks to complete their designs, motivated by large prize offerings, and provided with tutoring sessions and special assistance to learn the C# language[4].

We received 11 total submissions and checked each to ensure that the interface would allow the user to monitor the complete data set for the full 10 minute simulation period (as instructed). Unfortunately, a few of the interfaces had programming errors that prevented execution for the full 10 minute period. Others did not show the full data set as required. However, three interfaces did meet the design requirements and were deemed suitable for use in a usability study.

[3]http://www.expedia.com

[4]Special thanks to Tyler Newton for this expertise.

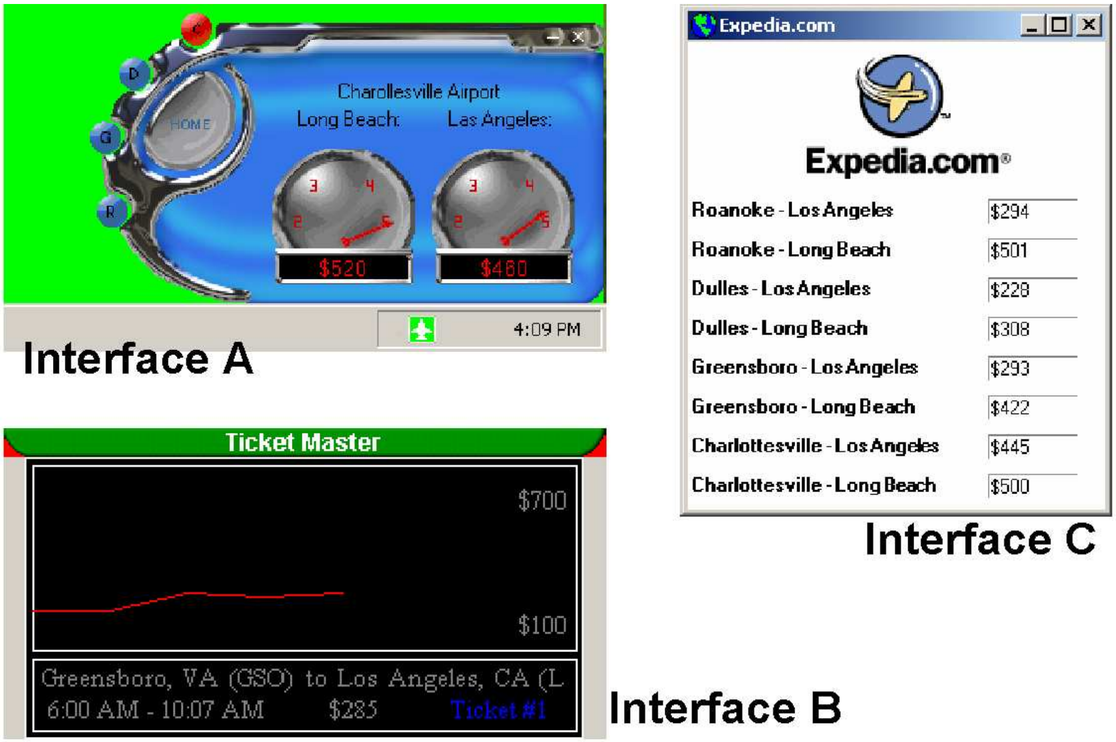


Figure 6.9: Notification system interfaces developed to keep users informed about Expedia.com airplane ticket prices, as studied in the case study usability evaluation.

Figure 6.9 includes a screenshot of each of the three systems. *Interface A* consisted of both a system tray icon that changed color to reflect price changes (green for price decreases and red for increases) and a oval-shaped space that included a dial indictor for each of the two destination airports. Buttons along the left side of the oval allowed the user to toggle between the four departure airports and view current prices—however, the system would continuously cycle through each of the departure airports without any user input. *Interface B* provided a graphical representation of ticket prices for a specific window of time. In its normal state, only the green bar at the top of the interface is visible, docked to an edge of the desktop. However, when the user clicks on the bar or when a price is updated, a graph for the particular flight appears for a few seconds. Users can manually cycle through graphs for each of the flights. *Interface C* simply displayed a list of the eight current ticket prices, updating each price with changes.

6.5.2 Testing and analysis procedure

To test the IRC-related usability of these three interfaces, we decided to use a IRCresults empirical platform. That is, we wanted to conduct a lab-based test of the interfaces with users and obtain the user's model IRCs based on actual performance data (as described in Section 6.3.3). This required building a platform[5] that could obtain the necessary performance data to calculate the concrete terms summarized in Table B.2. First, we state the initial objectives of the usability testing. Following this, we discuss the testing platform and test set-up, then describe the actual usability study and the results.

Test objectives. At the onset of the testing, we had never before used the test platform to collect real usability data, although the basic functionaly had been verified in several pilot tests. The tool itself and the test criteria (i.e., the price-buying rules for each round, questions used to probe comprehension of ticket price changes, etc.) had not previously pilot tested. The test itself (and availability of volunteer participants) was largely opportunistic, but was developed to meet several specific objectives:

- Of foremost importance, we wanted a proof-of-concept demonstration of the empirical IRCresults platform. By using the tool to conduct an actual usability comparison, the thought was that the development team and larger research group would gain confidence in the general approach. With the concept of this tool demonstrated fully, from theory to application, the groundwork would be in place for practical refinement, full development, and complete validation.
- To guide immediate development efforts, we were eager to collect test participant observations related to the adequacy of test platform screenflow, proctored and on-screen instructions, primary task calibration data for the sustainment calculation, other design decisions.
- As a final objective, we were curious (and hopeful) to see if the IRCresults empirical test platform might reveal differences between the three interfaces, according to any or all of the IRC parameters calculated based on actual user performance.

A platform for logging empirical IRCresults. To demonstrate the idea we had for an empirical testing version of IRCresults, we decided to convert the testing platform used in a few of the studies described in the background work of this research (Chapter 3). Specifically, we had conducted other lab-based testing of notification system artifacts with a block catching game as the primary task, but experienced frustration with the lack of a mechanism to accumulate and compare performance data. Our thought was that the IRC framework and the IRCresults equations should provide the comparison mechanism, if the automated testing platform was altered accordingly.

We recognized three general requirements for a platform:

1. Primary task performance data needs to be recorded at appropriate intervals (e.g., at least every few seconds), and the primary task must be of a nature to support fined-grained, meaningful performance data collection. This data must be characterized as having occurred in a solo-task or dual-task situation. At least some performance data in a solo-task situation needs to be collected so that sustainment can be calculated.

[5]Tyler Newton and John Booker assisted in this effort.

2. The platform must be able to characterize user performance related to signal response. A *signal* is a valued notification that should invoke a specific user response according to the user's goals. The platform needs to "know" when signals are presented, if and when users react to them (hits), and when the signal is no longer available. If reactions occur outside a valid signal interval, the platform needs to be able to recognize and record those reactions as false alarms.

3. The platform needs to be able to gather data to determine whether the user understood important information presented by the notification system. This should be accomplished by assessing the number of signals hit as opposed to missed, as well as by probing user recollection of information between rounds or in post-test questions.

Many dual-task testing platforms already meet all or portions of these requirements. However, when all requirements are met, the platform should be able to log and summarize the performance data necessary to calculate the user's model IRC. The general flow of participant activity within the testing platform was:

- the participant reads general instructions,
- the primary task is performed for a brief period of time to obtain a solo-task benchmark,
- the notification system and signal criteria are introduced and learned,
- the notification simulation is started along with the primary task,
- the participant responds to notification signals while maintaining primary task performance as appropriate,
- the simulation and the primary task stop,
- participants answer several questions about information presented by the notification systems,
- other post-test questions are completed,
- the process is repeated for other interfaces or other data sets.

In modifying our existing empirical test platform, we explicitly designed it to be reusable for testing notification systems. Therefore, we implemented the platform to execute from an external input file that specified key testing information, to include signal times and durations that would be simulated by the notification system, comprehension and projection information that users should have gained in the testing period, and general test administration items (instructions, screenshots, pre-test questions, the name of the notification system executable to launch, etc).

To obtain this information, especially the signal timings and comprehension items, we revisited the data set that the designers used in the creation of their prototypes (see Figure B.12). With the knowledge that each simulation would execute for 10 minutes, we settled on an appropriate number of signals to present during the testing session (about 5) and decided that no more than 20 percent of the simulation period should have an active signal present (this decision was thought to represent a very active period of Expedia price fluctuation). From here, we set signal criteria that users would learn and follow in the testing procedure. For example, users would be told to "buy" (or react to the notification system) if a price dropped below $300. In order to accommodate two rounds of testing, we decided to base each round of testing on flights to one of the two possible destinations (LAX or LGB). The testing criteria and resulting test input (signal timings and comprehension items) for the flights to LAX can be found in Figure B.13, while the same information for the flights to LGB can be found in Figure B.14.

The test platform does not interact at all with the notification system prototypes, except to launch and kill it as an external thread. To simplify recognition of signal and false alarm responses, we simply instructed the participants to press the spacebar if they thought they should buy a ticket or launch the full Expedia website for more information. However, participants were still able to interact with the simulated notification systems—they could cycle through screens and use any of the features implemented by the designer. This aspect of the testing platform made it readily interchangeable with any notification system prototype, but still able to process user input.

With the test information established and in an appropriate input file, we only needed to add a few lines of code to summarize performance data as concrete IRC terms and then use these concrete terms and the IRCresults equations to calculate a user's model IRC. The platform logged all results as an output file.

Usability study procedures and results. Eighteen undergraduate computer science students participated in our usability study for class credit. Each participant completed two rounds of testing with a different interface. Round 1 always tested with the LAX portion of the data set (although notification system prototypes also displayed the LGB data as well), and participants were quizzed on the criteria for buying tickets according to Figure B.13. Likewise, round 2 always tested with the LGB portion of the data set and the criteria in Figure B.14. Each participant used an identical desktop computer in a controlled lab testing facility.

The only concrete IRCresults term not calculated through participant performance data is cost of interruption (COI), which is more of a situational variable. We set COI at .33 for this study and ensured that this value was generally reflected in the role-playing instructions provided to the participants. In the verbal and written test instructions, participants were asked to think of themselves as an administrative assistant in a local company, who's employees frequently travel between Los Angeles, CA and Blacksburg, VA throughout the week. As part of their job, they make travel arrangements for other employees, to include the purchasing of airplane tickets. To ensure that the company saves as much money as possible, they are asked to actively monitor prices and purchase tickets when the price trends are favorable–at least one of the other employees will be able to use the ticket, but purchase prices are reviewed closely by the accounting department. However, they also have many other important tasks they perform throughout the day (simulated by the block catching game task), so they cannot devote full attention to ticket monitoring.

The raw data and user's model IRCs generated from the first round of testing are depicted in Figure B.15. Before the test, the design model IRC (or "Target IRC" in the figure) was set with the IRCspec tool according to general requirements conveyed in the design specification. A highlighting scheme is used to convey the cases where a user's model parameter matches within $\pm$ .10 of the design model (green highlight), $\pm$ .20 (yellow highlight), or more than $\pm$ .20 (pink highlight). Of course, if a design caused a user to have the exact level of psychological impacts that were expected by the designer, the design model–user's model parameter disparity would be $\pm$ 0.

From the user's model IRC values and matching trends, it is apparent that all three interfaces provided appropriate levels of interruption (although perhaps a bit too low), but generally failed to support goals related to moderate reaction and high comprehension. However, it was encouraging that for Interfaces B and C, at least one of the six participants was able to use the notification system as it was intended.

Identifying usability problems. With these high-level IRC summaries of empirical testing results, we can readily distinguish designs that meet intended psychological effects from those that do not. However, in order to speculate and make claims about *specific causes* or usability problems that underlie abstract design model–user's model parameter disparities, we must return back to the concrete terms obtained in the study.

For instance, an evaluator may wish to gain insight into why the designs failed to support targeted levels of reaction (reaction values obtained were approximately -.30 below the design model for all three systems). We can use the R equation to identify concrete value changes (hit rate and response rate) that would have the desired impact.

Reflecting on the results for Interface B, improving the already-strong response rate (average = .77) would have little effect. However, if the average participant responded to three of five signals (rather than just one), the hit rate would be high enough to yield design model–user's model agreement (-.04 difference). Recalling that a signal in this experiment was defined as a ticket price dropping below a pre-defined threshold, the designer should consider task characteristics and information design aspects that contribute to poor hit rate performance. For example, design observations that an evaluator might make to explain the lower than desired hit rate might include:

- The exact ticket price is provided in the bottom-center of the interface, BUT it does not stand out from other information in that area. A highlight or contrasting text color could make the exact ticket price more distinguishable.
- The ticket price trend is shown by the line graph, BUT the vertical scale lacks the precision to determine when specific pricing thresholds have been crossed. A finer scale should be designated with tick marks.
- Users are expected to buy a ticket when prices drop below a pre-determined threshold, BUT they may forget what that threshold is and the interface does not provide any visual reminder.

> If the user is able to specify the buy-threshold, the threshold can be depicted in the price line graph or an exact price below the threshold can be given a highlight color.

Since the hit rate only needs moderate improvement, only a slight change to the interface may be needed (possibly ruling out the third point above). However, the evaluator also should consider the other concrete terms of concern. For instance, as the perception rate (ratio of hits to total responses, a factor in the comprehension parameter) is only .33 on average for Interface B, the design flaw is likely to be related to users not distinguishing between hits and false alarms. Therefore, a more elaborate design strategy may be the best approach, perhaps the suggestions following the third point above.

In this pilot study, we only intended to demonstrate how an evaluator might go about identifying causes of the usability deficiencies or attempt to reengineer the designs to improve the results in the phase of testing. Future work involving iterative design improvement and empirical IRC reevaluation would be necessary to validate the technique. More importantly at this stage, we wanted to explore the requirements relating to the creation and use of an empirical IRCresults testing platform that would be necessary for technique validation. Therefore, having conducted and presented a pilot test procedure, we are able to reflect on the broader implications of such a tool and improvements that can be made in its continuing development. Although we shift focus to this reflection here, the empirical version of IRCresults is revisited in the next chapter.

6.5.3 Broader implications: A full spectrum of usability data

This initial user study with the empirical IRCresults tool shows much promise for its continued development. Using the tool to conduct a usability study with a large number of participants allows a wealth of usability data to be collected at various levels of abstraction. At the lowest level, all participant events and comments have been logged in a text file and are accessible in full form to probe very specific questions. At a higher level of abstraction, we can examine performance related to specific signal events or segments of a simulation for the test sample at large. Although we calculated a single IRC value for the entire round, we could have calculated several intermediate IRCs for various segments of interest. Of course, at the highest levels of abstraction, we can look at individual IRC values for test participants or consider the overall average. In this way, we are able to characterize specific event sequences, artifacts, performance by specific participant demographics, or overall system tendencies.

Our approach of having a standard data set on which prototypes are developed, as well as a reusable and highly modular testing platform (run from a script file), certainly encourages replicable and comparable testing procedures. This approach provides the groundwork for reference task research methods within HCI, a proposal that is more fully explored in the next chapter.

6.6 Conclusions

This work provides another step toward a long-term proposal for integrating critical parameters, mediated evaluation, and claims reuse in interactive system design and evaluation activities. Our sincere hope is that the analysis and potential approaches we suggest will continue the dialog on methodological and practical aspects applicable to notification systems. Though rooted in the study of notification systems, we feel that our research approach can generalize to other classes of systems.

Claim 8 – The IRCresults equations and tools…

Broader Implications, **Sections 6.4.3, 6.5.3**

+ demonstrates sufficient interrater reliability with an analytical assessment tool

+ expresses results that are meaningful to system nuances, helping analysts detect usability issues

+ enables design comparison and artifact reuse

+ compatible with a broad spectrum of usability data abstraction levels

− using the tools may not occur to a designer, since the tools are not a ready part of their work environment

We have mentioned directions for future work throughout this chapter. Specific contributions of this effort are:

- Summary of arguments for and against using critical parameters to characterize user goals and usability artifacts,
- Variation to the concept of a critical parameter, which would allow benefits related to both abstract and concrete knowledge representation (see Table B.2),
- Equations and usability evaluation support to elaborate and allow evolution of notification system critical parameters,
- Case study illustrations of how general instruments (analytical and empirical) allowed meaningful comparison of system designs, resulted in valuable inferences for reengineering, and can provide a full spectrum of replicable and comparable usability data.

Certainly, future work will improve our understanding of the critical parameters and the concrete variables that instantiate them. Just as with other scientific disciplines, we can use equations as a language for expressing general relationships between difficult, context-sensitive ideas. With the equations as a point of convergence for a suite of testing methods and tools (referred to as

IRCresults), we can develop instruments that are appropriate to the situational and task context, but allow comparison and facilitate reuse across designs. An important next direction will be to mitigate the downsides expressed in Claim 8, by finding ways to make the tools more accessible to designers.

As noted earlier, our approach makes explicit many aspects of design that researchers are sometimes uncomfortable with. For instance, the notion of setting a user goal and psychological effect like reaction to a linear axis often evokes resistance. However, we suggest that the notion can be embraced as a conceptual metaphor and tool for dialog. We believe that extending the idea of critical parameters [97] and conceptual models [100] from original intentions may inspire improved methods for HCI research.

Primary benefits of this approach may be found in helping novice designers reflect on design tradeoffs and conduct mediated evaluation. Concepts articulated by equations, tools, and visualizations improve the chance that these designers will be intrigued by HCI problems and develop innovative solutions. We are also hopeful that a critical parameter approach to interactive design research dialog can improve consensus of key issues, comparison of new efforts to existing efforts, and development of context-specific usability testing methods and instruments. As the community looks for approaches that will increase the likelihood of science of design, or support the practice of usability engineering, these arguments should be of interest, broadening as a topic of continued debate.

Chapter 7

REUSABLE DESIGN KNOWLEDGE WITH CRITICAL PARAMETERS

In previous chapters, we have developed an argument that the IRC framework can provide accurate and consistent characterizations of the notification system design model and user's model. This implies that designers and evaluators can reliably use the IRC to describe intended interactions and guide formative evaluation efforts. However, we have a broader vision regarding the potential impact on a notification system's usability engineering process. We speculate that the IRC framework may improve early-stage notification system prototype development by facilitating identification and application of basic research (guidelines, patterns, and theories), enabling cross-domain design knowledge reuse, guiding an iterative interface design cycle, and providing benchmarks for evaluating progress between design iterations.

A general, long-term research question that emerges is:

Does the IRC framework have potential to serve as a research framework that would allow design improvement over time?

Our thought is that we can make progress toward this, if the framework can provide an access point to a design library of components and claims. To begin investigating the question, we describe in this chapter the steps taken to initialize a claims library and integrate the IRC framework throughout a tool-supported design cycle.

These efforts are not intended provide definitive validation or answer all the questions that surround our proposed approach toward managing usability engineering knowledge and process. On the contrary, this portion of the work is intended to lay the groundwork necessary for broader dissemination and long-term study of the IRCspec and IRCresults tools–opening many new questions that will frame several years of future work. A key objective of this process is to extend the underlying concepts of the IRC framework to meet the design knowledge reuse themes put forth by Carroll, Sutcliffe, and others. This chapter may be thought of as bridge from chapters 5 and 6 (IRC tools for system description and usability evaluation) to chapter 8, where we demonstrate how the body of work connects together and leads to new paths for future research. Therefore, much of the work reported here is synthetic and exploratory (though largely published in a diversity of forums).

The first section (7.1) describes the progress that we have made with developing and evaluating a claims library, progress which has been reported in two conference papers [101] (appeared at the IEEE International Conference on Information Reuse and Integration) and [49] (appeared at the ACM Southeast Conference). Through this work, we show how the tenets of Domain Theory [118], which allow access to reusable design knowledge, can be extended with a critical parameter-based framework like the IRC. Through the blending of these concepts, we are able to elaborate on the idea of problem frames, as shown in the next chapter.

Section 7.2 discusses the ongoing efforts to integrate the IRC framework with the claims library through a suite of design-support tools (LINK-UP). LINK-UP is an integrated design environment that operationalizes Carroll et al.'s concept of claim analysis [28]. This effort was undertaken to mitigate Claim 8 in the previous chapter, situating the IRC tools so that they are more accessible to designers during design work. Research products resulting from this effort include published papers in the *Proceedings of the Conference on Computer-Aided Design of User Interfaces (CADUI '04)* [32], in *Proceedings of the World Conference on Educational Multimedia/Hypermedia and Educational Technologies (EDMEDIA '03)* [35], a paper at the conference on Software Engineering and Knowledge Engineering (SEKE '04) [129], a chapter for a book on cognitive systems for software design [86], and papers at the Participatory Design Conference 2004 [94] and the E-Learn 2004 conference [75]. As discussed in the next chapter, the progress made through this project enables exciting directions for future work–deeper and more situated investigations of design knowledge repositories and approaches to a *science of design*, improvement of HCI educational materials, integration of multidisciplinary design perspectives, and implementation of a reference-task agenda for notification systems.

7.1 Initializing a Claims Library

One of the key benefits in being able to represent critical parameter design models and user's models includes storing and accessing claims. A catalog of claims can provide new design ideas, assist in comparing alternatives, and help designers identify important system tradeoffs in early development phases. While designing retrieval mechanisms for claims can be a daunting task, our thought is to use design model and user's model IRCs to aid classification and retrieval strategies. This section describes our initial efforts to create a claims library of notification design artifacts that illustrates this notion.

7.1.1 Motivation

Recent work within HCI has focused on developing theories and methods for design reuse in the requirements generation stage [118, 24]. As part of this work, the Domain Theory provides a structure of abstracted domains, interaction sequences, and tasks that can be used to catalog design information. Domain Theory provides a roadmap that is extendible to any design domain–obviously, our concern continues to be for notification systems design. As a key part of scenario-based design [108] and extension of Domain Theory, *claims* provide a strong basis for reuse, since they are concise statements about the benefits and limitations of an artifact in use. However, claims must be "factored" (a term developed by Sutcliffe), or made generic in order to achieve a sufficiently broad level of abstraction to be frequently accessed [116]. The drawback of this process is that factored claims may be stripped of important contextual information, potentially allowing inappropriate claims to be presented as design advice.

Claim 9 – A claims library can be used by designers in early development stages...

+ provides access to new and alternate design ideas

+ helps designers recognize key SBD tradeoffs, discovered through basic research and previous development efforts

+ Sutcliffe's Domain Theory provides a roadmap for claims library development

− BUT, uncertain how Domain Theory alone can express critical dual-task user goals and effects → Claims Library, **Section 7.1.3**

− BUT, no claims library or content available for testing → Initial Studies, **Section 7.1.2**

Although we are excited about the notion of a claims library for notification systems design, we do not believe that context of the dual-task situation and user tasks can be easily maintained with the 14-item claim format, as suggested in [118]. While some of these important variables can be indirectly embedded in associated Domain Theory objects and tasks, scenarios, or even keywords, we would like to preserve direct characterization of the attention-utility theme and abstraction of critical user goals (discussed in Chapter 4). To this end, we believe that user's model IRC ratings (or more generally, any design space characterized by critical parameters), if enabled as a primary index and used with other Domain Theory tenets, can provide an improved context for claim reuse.

7.1.2 General approach and expected results

General approach. There are several parts to this overall project, an initialization of a notification systems claims library founded on both Domain Theory and the IRC framework. We approached this project as typical iterative system development effort.

The first iteration was begun as a class group project[1] under the premise of evaluating the suitability of Domain Theory for organizing the reuse facility for applications within a particular

[1]CS6724, Spring 2003, Design and Software Reuse for HCI. Other group members were: Dave Myers, Con Rodi, Catherine Payne, Chris Allgood, Chuck Holbrook, Andrew Ray, and Sourabh Pawar

class. First, as part of the requirements analysis, we considered the elements of Domain Theory that provide the most likely anchor points for existing notification system design. Then, we developed the core system architecture and user interface to enable a web-accessible relational database. Next, we began the long process of converting a wide variety of existing notification system design knowledge to an appropriate data definition that includes essential claim information, IRC classifications, and generic tasks. We evaluated these efforts with an internal heuristic evaluation and a lab-based, formative user test. Details on these efforts are covered in the first parts of Sections 7.1.3 and 7.1.4.

Encouraged by feasibility of the initial iteration, we continued development of the claims library beyond the class project. With the assistance of undergraduate programmers[2] and researchers[3], we extended core features of the library to include mechanisms that allow easy entry and editing, administrator rating, and browsing of components (working with the IRCspec and IRCresults systems). These additional development efforts are discussed in Section 7.1.3. During this stage, we specifically considered design options that would help library contributors specify IRC and generic task metadata. To inform this process, we conducted additional user testing that is described in Section 7.1.4.

Expected results. While this had potential to become a very large project, we expected that a minimally populated demonstration system and initial feedback from potential users would be sufficient proof-of-concept. We believed that this would demonstrate the utility of the IRC framework, or another system class design space defined by critical parameters, in supporting design reuse. An enduring contribution would be the claims library itself, which can serve as a web repository to which designers can add new designs and components, as well as usability data and claim annotations. This could be useful for supporting seminar and class projects, and provide an interesting base for future collaboration and investigation.

7.1.3 Developing the Notification Claims Library

To establish our claims library for notification systems, our first step was to gather content from a wide variety of systems and consider the library system requirements implied by activities inherent in cataloging this content. We considered approximately 50 systems, to include ones that were designed within our research group, through research collaboration with other designers, and systems from related literature. We intentionally selected systems that included a variety of implementation platforms and supported a range of user tasks.

Notification Collage is an example of one of the systems considered [56]. This interface is intended for a large-screen display that would be situated within a common work area. The system provides a bulletin-board type forum for delivering information regarding the status of personal contacts and ongoing events. In continuing to describe our content creation process in the

[2]Edwin Bachetti and David Felton

[3]Members of the Fall 2003 Notification Systems Seminar: Alan Fabian, Melissa Grant, Cyril Montabert, Kevin Pious, Nima Rashidi, Anderson Ray Tarpley III, and Nicholas Taylor

paragraphs below, we revisit a claim generated from this system. As we collected content for our library, we encountered two major challenges:

- The first challenge stems from the tension inherent in the claim generalization process itself, trying to ensure that a claim contains meaningful design knowledge, while maintaining a level of abstraction that allows access to the claim via the library's information retrieval mechanism. Preserving this balance proved to be difficult.
- The second challenge was determining the precise nature and form of the abstraction layer, which provides the structure for the information retrieval process.

Once resolved (as detailed below), we were able to develop an effective classification strategy and validate that the library was useful through initial user testing.

Domain Theory anchor points. To address our first challenge, we needed to adapt the Domain Theory abstraction scheme to be more representative of notification system design concerns, specifically focusing on design abstractions from the design metadomain and use of generalized and generic tasks [118]. We began with a claims format based on the 14-point claim recommended by Sutcliffe [116]. This format is ideal for collecting not only the important knowledge related to the theoretical grounding of the claim, but also the articulation of the design tradeoffs and context. In addition, this format allows for documentation of claim history.

As we considered this format in relation to its purpose in capturing knowledge about notification systems, we realized that slight adjustments would need to be made. The more important adjustments related to the definitions of pre-existing fields. For instance, since we were accustomed to representing the effects of a notification system with IRC ratings, using the Effect (9)[4] field seemed like the logical place to record this measurement. Similarly, Sutcliffe intended the Scope (14) field to provide relation to Domain Theory components, and we anticipated that this could be different for the notification system's primary and notification tasks. As a result, we broke this field into two sub-fields. These fields were originally intended to be the primary indexing mechanisms for the claims library.

We feel that the users of our knowledge repository will value the Upsides (6) and Downsides (7) of the claim, the Scenario (8) from which these tradeoffs originate, the related usage issues (10, 11), and theoretical background (12). We augmented this with the addition of the Parent Component (15) field to provide contextual reference to the system that implements the designed artifact.

Fundamental information from a maintenance perspective includes the Claim ID (1) and Author (3a) fields. Less obvious is the Relationship (13) field that further establishes the context of the claim according to other notification design issues. We anticipate that designers will be able to add most of the claim data themselves; however for quality control and index verification, we saw the need for an Editor (3b). Claims are intended to evolve as a design is iteratively developed and

[4]parenthesized numbers refer to the fields depicted in Figure 7.2

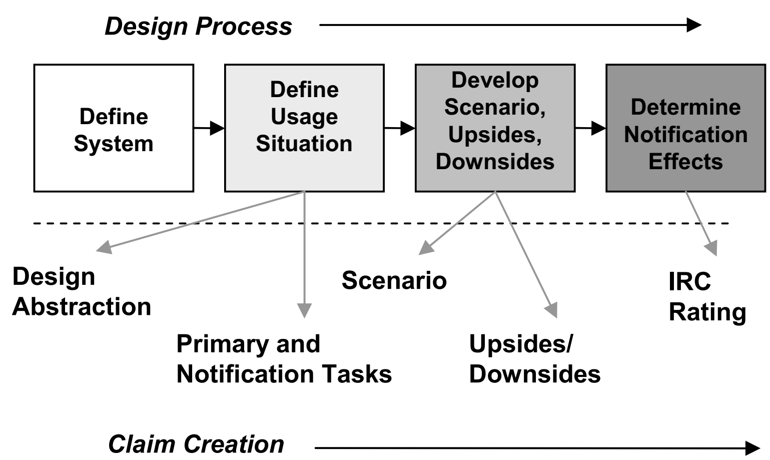


Figure 7.1: The process we followed to establish claim content, describing design knowledge of a notification system artifact.

evaluated in actual use, so it seemed logical to include a Usage Log (16) to capture these changes. Figure C.1 shows the modified claim format, indicating our changes.

Generic and generalized tasks. Once a claim was created in this format, our next step involved generalizing this claim to an abstract layer. Our initial definition of this layer was simply a generalized version of the original claim, similar to examples provided by Sutcliffe. Our abstract claim contained (1) primary and notification tasks, (2) IRC ratings, and (3) upsides and downsides. We planned to create a relationship between the specific claims and the abstract claims on the basis of their primary tasks, notification tasks, and IRC value. To create this relationship, we needed to describe the tasks in a way that allowed them to be grouped in different categories, which would then be captured at the abstract level.

We accomplished this by using a discrete vocabulary derived from the generic and generalized task models provided by the Domain Theory[5]. The two models differ on the basis of task complexity. Generic tasks describe activity at a much more granular level, making them suitable to describe the notification tasks. As primary tasks tend to be broader in the range of activity that they encompass, the more complex generalized tasks (compositions of multiple generic tasks) were used. Figure 7.1 shows how the claims creation process mirrors the design process.

Claim creation process steps. The claim creation process is as follows:

- **Defining the system and usage situation.** The first step was to identify a system from which we wanted to capture usability experience, like Notification Collage. Next, we identified the usage situation in which we wanted to evaluate the system. Here we decided to look at the

[5]Generic and generalized tasks are listed and defined at $http : //www.co.umist.ac.uk/hci_{d}esign/appb.htm$

live video feed capability of Notification Collage in the case where the feed supports monitoring the status of another person. However, the key notification system design challenge would be that this monitoring would occur while the user is simultaneously engaged in other office work. We also identified a claim title and chose the key word(s) to describe the primary and notification tasks, beginning the development of a claim record (see Figure 7.2a). In this instance, we chose monitor, interpret, and evaluate as notification tasks. The user has goals of maintaining awareness about the availability status of another person (monitor), and based on his/her understanding of this status (interpret), determines a suitable time to interact (evaluate). We chose planning-scheduling as the primary task because the user will incorporate their understanding of the other person's availability to sequence collaborative and non-collaborative work activities.

- **Selecting the Design Abstraction.** Next, we identified the general design technique (Design Abstractions in Figure 7.1) that characterizes how the system or system component interacts with the user.

- **Developing the scenarios and tradeoffs.** Next we develop the scenario, which is used to ground the usage context of the system [108]. To continue the example with Notification Collage, we created a scenario to describe a typical usage situation. From this scenario, we identified the pros and cons of the design, which are articulated in the upsides and downsides for the claim. The description summarizes the designed artifact in the context of use, as shown in Figure 7.2b.

- **Determining notification effects.** Having established the review of the design feature in the context of its scenario of use, we turn our attention to the notification effects. After establishing this rating, we make note of any relevant dependencies, issues, or theories that should be considered during notification system design. Figure 7.2c continues our example, showing the completion of all non-trivial claim fields.

Cataloging the claims. During this claims creation process, we indirectly established a classification system or indexing mechanism for the catalog. However, this process deviated considerably from the method Sutcliffe proposes. We describe our experiences, which ultimately lead to our recommendations regarding the design of a claims library.

We initially encountered a problem with articulating the primary task for our usage cases. Some of our systems were very ubiquitous, which made identifying a primary task in the general case practically impossible. This in turn meant that we could not selectively map the specific claims to the abstract claims, as Sutcliffe and Carroll's factoring process intends [116]. We solved this problem by not focusing on what the notification system can support, rather on a specific primary task goal. This direction allowed us to be able to identify a discrete primary task.

By increasing our dependence on the scenario, however, we raised a new issue with respect to abstraction. Our claim became even more reliant on context, which is contrary to the principle of a.

2. Title	Video Feed for Status Monitoring
4. Artifact (Design Abstraction)	Video
14a. Scope (Primary Task)	Planning-Scheduling
14b. Scope (Notification Task)	Monitor, Interpret, Evaluate
15. Parent Component	Notification Collage

b.

5. Description	Using an extraneous video feed displayed on a large screen display to monitor the presence of a remote individual.
6. Upsides	+ Live video allows a quick and easy way of showing presence, + Posting of live video, sticky notes, slide shows, etc, affords a wide variety of media forms, + Lack of audio decreases interruption and information overload, avoiding sensory overload, + Customizing the rate at which the video feed updates allows the user to control interruption.
7. Downsides	– Live video broadcast reduces privacy for users, – User-initiated interruption is hindered because users don't have the ability to control the refresh rate on a video feed, – The user can miss a change of status because of lack of audio or other non-visual cues.
8. Scenarios	*Bob is working with Alice on some paperwork in their lab. Bob must frequently go over to Alice's workstation to get her to look at the paperwork and sign documents that are needed. Alice, however, is often not at her workstation. Bob can fire up the Notification Collage (NC) on a second monitor and ask Alice to post a video feed of her workstation area. Bob will be able to continue doing his work, but he is aware of Alice's presence. By glancing at the NC when Bob takes a break or has a need to talk to Alice, he will be able to quickly realize if Alice is present at her workstation. He will no longer waste time walking across the lab to find Alice.*

c.

9. Effect (IRC Rating)	Interruption = 0.2, Reaction = 1.0, Comprehension = 0.9
10. Dependencies	Transmission bandwidth, resolution. The refresh rate (presumed at every 5 min, for some privacy preservation and lower bandwidth transmission) prevents monitoring of the display during natural primary task break points.
11. Issues	Privacy of monitored person
12. Theory	Ackerman's social-technical gap (privacy); McFarlane's user-initiated interruption (refresh rate); Dourish & Greenberg (awareness for reciprocity and privacy)

Figure 7.2: An example of the claim record development, emerging in steps that followed the claim creation process: a) describing design knowledge of a notification system artifact; b) sample claim description, scenario, upsides, downsides; c) claim fields relating to consideration of notification effects, pertaining to the Notification Collage example.

abstraction and basically removed the property of domain extensibility from our abstract claims. To resolve this dilemma, we decided to change our original definition of an abstract claim. Rather than having an abstract claim exist as a generic, higher-level variation of a specific claim, we decided to focus on functionality.

Instead of abstract claims, we used a “footprint” approach, recognizing that each specific claim exists in an abstract, four-dimensional space based on its primary and notification tasks (Scope) and two new indices: IRC ratings (Effect) and Design Abstraction (Artifact). Because the primary and notification tasks were based on the generic and generalized task models described in the Domain Theory, they already placed the claim on an abstract level. The Design Abstraction attribute is a generalized way of describing how a system interacts with a user (e.g. color, animation, or audio). IRC ratings describe the effects that a system will have on the user (e.g. causing high levels of interruption) in non-specific terms. A claim’s existence in this four-dimensional space constitutes library abstraction.

Continuing our Notification Collage example, we look at the four variables used to place the claim in the library. Since the primary and notification tasks were already in the vocabulary of the tasking models, they are ready to be mapped, as is the IRC. The final variable needed to complete the abstraction of our claim is the Design Abstraction variable. To find this, we considered possible Design Abstractions and chose those that best described the component of the system we were evaluating. In this case, we chose Video as our design abstraction (see Figure 7.2a). Once we had identified these four variables, the claim could then be included in the library. Note that in this case, there are three notification tasks, so there will be three different combinations of these variables. The union of all combinations is the footprint. If a user’s query falls anywhere in that footprint, the claim is returned.

With this cataloging scheme in place, we turned our efforts to building tools that would allow design knowledge to be input and accessed in the claims library.

Tools for converting design knowledge to claims. A key priority was having the claims library web-accessible and highly modular to facilitate continuing development efforts, so we decided to implement with Java servlets and JSP, using hibernate as a JSP-to-XML translator, on a Tomcat/Apache server. Through the class project effort, we developed a relational database structure (implemented in MySQL) to support basic searches for claims, which allowed formative user testing proceed. Searching was implemented to support keyword matching as well as faceted search based on the indices defining the claim footprint (IRC values, general primary and notification tasks, and the design abstraction), see Figure 7.3. As a parallel series of efforts[6] we developed other tools that would allow users to add and edit claims, sequentially browse through the existing claims, and to rate or grade the quality of a given claim.

Figure C.2 shows a prototype of a claim within the claims library, as it would appear to the user. This prototype reflects several key decisions. First, the multimedia components (e.g.,

[6]Conducted beyond the class project, with the assistance of two undergraduate programmers, Edwin Bachetti and David Felton

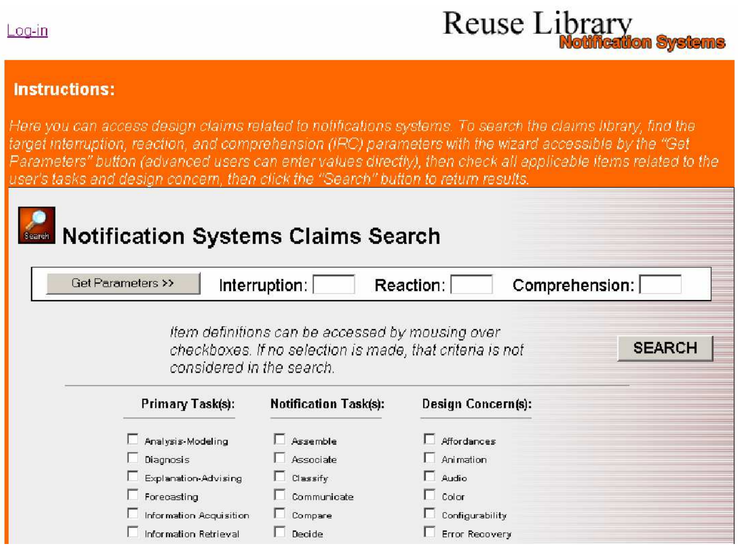


Figure 7.3: Screenshot of the faceted information retrieval interface for the notification system claims library.

screenshots or brief movies of the artifact, executable versions of the system containing the artifact, etc.) were thought to be quite valuable to users attempting to get a sense of what the artifact actually is. Second, since the claim results from important nuances of both a primary task and a notification task, we decided that the scenario of use must be as clear as possible. Since the IRC rating stored with the claim is elaborated in the scenario, and the user would find the claim based on a match of the IRC rating, supported tasks, and/or the design abstraction, a clear understanding of the scenario would be required to determine whether claim applies to a situation that is similar to the one in question. Therefore, we envision a scenario narrative that is illustrated with additional multimedia components, such as sketches or movies showing user interaction. Finally, we received many suggestions from potential users requesting that a record of user comments about the claim and an overall claim quality rating be made accessible with the claim record. The claim viewing pages were implemented according to this prototype.

To order to support addition of claims into the library in a user-friendly manner, we developed a series of screens that would allow a user to input the necessary information (depicted in Figure C.3). This interaction follows the claims creation process we described earlier (see Figure 7.1) and allows the fields of the claim format to be specified (see Figure C.1). Each input field has a link to example content and a listing of key considerations. Primary and notification tasks are selected through the use of checkboxes, and the anticipated IRC values (design model) can be manually entered or calculated with the IRCspec tool. The interface allows uploading of multimedia objects to depict the artifact as clearly as possible. Since the data is stored in a relational format, several claims can be made about a given artifact, several artifacts can be described by a given scenario, etc. Similar to the claims adding tool, we developed features to allow editing of claims (see Figure C.4).

Another initial set of features allows an administrator to establish a rating of claim quality, expressed by up to four stars (i.e. a claim rated at "one star" is of poor quality). Using input provided by seven potential claims library adminstrators[7], we determined that the claim rating should be based on five factors:

- **Richness of artifact description**, subjectively graded by an administrator after the claim is added to the library (e.g., having a movie of the artifact within the claim record would generally result in a higher score than a single screenshot).
- **Technical merit of the theory grounding the claim tradeoffs**, judged by an administrator based on academic merit of the literature cited in the Theory field of the claim record (e.g. if the tradeoffs express an idea that appeared in an HCI or psychology journal, the claim would be scored higher that if the tradeoff was not supported by any published material).
- **Experience level of the claim author**, assigned within the claim author's user profile by an administrator (e.g., if the claim was contributed by an undergraduate in the Introduction to HCI class, it would receive a lower score than one contributed by an experienced notification systems developer).
- **Usage frequency of the claim**. As users select a claim to be used within their projects (the idea of projects is fully discussed in Section 7.2), a counter is incremented for that claim (e.g., as claims are used in more projects, their ratings increase).
- **User opinions** about a given claim can be expressed by users as they browse claim library contents and view a claim record. Comments can be added, a a vote regarding the quality of the claim can be cast. As the average user opinion about a claim increases, the overall claim rating increases as well.

We implemented a screen to allow administrators to view the claims that have been added to the library, but not have not yet been assigned a rating (see Figure C.5). This screen allows the administrator to view appropriate claim fields (i.e. the theory field and tradeoffs) and enter scores for the first two factors. The final three factors are automatically calculated and keep updated by the system. The current overall rating of the claim is displayed at the top of the claim record.

Claims library content development. With tools in place to allow addition of claim content to the library, we have established a large enough collection of claims to permit user studies and provide proof-of-concept. Future work will broaden this collection significantly, with a special effort involving multidisciplinary researchers; however, we briefly describe the current state of the claims content–totalling about 130 claims. Our initial efforts attempted to establish a fairly broad coverage of each of the four indices, but develop some depth for a few specific index combinations. That is, we wanted to include claims that address almost all of the generic and generalized

[7] Students in the Fall 2003 Notification Systems Seminar

tasks, design abstractions, and IRC corners. However, a specific index combination like "indicators (IRC is close to 0/1/0) that support monitoring with animation" would be deepened with at several different claims. To support efforts to demonstrate cross-domain and cross-platform claim comparison, we generated claims from more than 40 different notification systems.

In addition to our effort at establishing claim breadth and points of claim depth, we also established two collections of claims relating to a specific design problem, but depicting a variety of design options. The first collection (referred to as the *CNN collection*) relates to notification systems that could be used to deliver summaries or "text + graphical" information about changing news headlines on a news site like CNN.com. The second collection (referred to as the *Expedia collection*) relates to notification artifacts that convey changes in Expedia.com ticket price information. Project resources are also available to easily develop a third collection of claims relating to alternate implementations of the Scope notification system interface. Discussion of the use of these collections is included in Section 8.3.3.

A final effort in content collection demonstrates the conversion of empirical testing results, such as those included in Chapter 3, to claim format. Having the IRC framework, the IRCresults equations, and the claims library allowed the results from each of the four empirical studies presented in Chapter 3 to be archived in a reusable claim format. With sufficient claim content developed in the claims library, we were able to design and execute initial user testing, described in the next section. [8]

7.1.4 Formative testing

Testing was designed to be a formative, proof of concept demonstration that establishes how well the current library design, organization, and content supports notification system requirements reuse. Through this exercise, we were particularly interested to learn whether a critical parameter classification system, like the IRC framework, can serve as an effective indexing complement to Domain Theory for a claims library.

User study #1: Faceted vs. keyword search. As reported in [101], after developing the system architecture and claims catalog user interface, we were eager to receive feedback about how well it supports the design for reuse process. We wanted to get feedback on how well our abstraction method supported information retrieval, so we had nine participants use the system by trying to apply the populated library in a series of design tasks–five of which used the faceted search scheme (consisting of IRC, primary task, notification task, and design concern specification) and four who used a standard keyword search. While the design tasks tested many basic aspects of the entire library, we focused on how useful the participants comparatively felt about the effectiveness and intuitiveness of the claim search mechanisms and retrieval process. We were primarily interested to

[8] The claims library database is available at http://research.cs.vt.edu/ns/research/Christa/claimsDB.sql. This database includes real claims (claim IDs 8 to 53 and greater than 152182) as well as the 150,000+ automatically generated claims. The database tables include other information used to implement the LINK-UP system, but the "claim" table and immediate relations store the claims.

see if library users were able to negotiate the retrieval process through the generalized and generic tasks, after abstracting the given design tasks.

The participant interaction with the library was as follows. A design problem was presented and the participant was asked to search the library for claims that would help them most with the task. They would then create a query with the search mechanism they were given. Based on the query, a number of claim summaries were returned. If the participants wanted more information on a claim, they could expand it to view the entire claim. Participants would select the claim if it was determined to be helpful to the design task.

After the participants finished the series of four design tasks, we asked them for feedback on whether the tool helped them locate relevant design knowledge, how effective the library was in doing so, and what the library was most useful for. Roughly half of all participants (equally divided between search conditions) felt the library was useful, and the majority overall felt it was most useful in "inspiring new ideas over reinforcing design assumptions." With respect to how effective the library was in "helping out with the design task," four said it was effective for all scenarios, four said it was effective to some degree in all scenarios and one said it was effective for some scenarios. Based on these comments, we feel our claims content and format was successful in terms of storing design data.

While it was encouraging that participants thought the claims library was generally useful, we were disappointed that, between participants using the two search mechanisms (faceted or keyword), there was little apparent difference in attitudes and only a slight difference in claim selection performance (with only two participants that used the faceted search mechanism performing much better than the four keyword searchers).

While the majority of the faceted search participants felt they were able to match the search terms to the given scenario, some felt that the terms were too vague and that there were too many choices. The observation about vagueness is indicative of the tension in finding the right balance of generality and specificity with respect to abstraction. Our follow on study attempted to more deeply probe the user reaction related to the faceted search support.

User study #2: Comparing potential indices. To improve our understanding about the difficulty novice users of the claims library might have with constructing searches to find claims that address a notification systems design problem, we asked a group of undergraduate notification systems researchers to probe this question. Together, we planned a lab-based user study to identify specific portions of the faceted search page that students have problems comprehending.

In preliminary testing, the research team determined that selecting a primary task from the generalized tasks list was nearly impossible. Even when assisted in an interactive discussion session, three of the seven undergraduate computer science students were unable to agree that a primary task like "surfing the web" could be generalized to "information acquisition" (one of the generalized tasks); six of seven were certain that they would not have made that association on their own. However, they had few problems associating the notification tasks embedded in an interaction scenario with the generic tasks. Based on this, we allowed the students to replace the

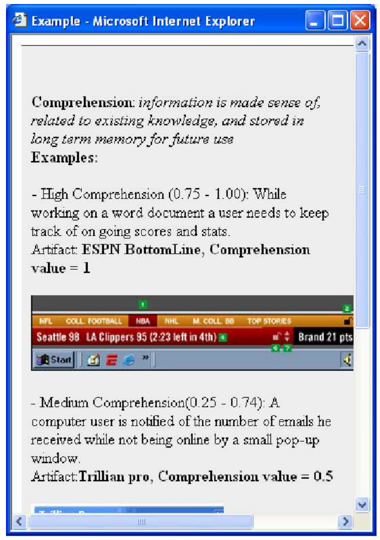


Figure 7.4: Example of the help screen features added to the faceted search page for the second user study.

primary task selection step in later test sessions with selection of the general interface platform or environment (i.e. desktop, virtual environment, mobile device, etc).

At this point, we had already validated that the IRCspec tool could allow users to specify consistent design model IRC values; however, our research team was interested to see if the search screen design should enforce the use of 16+ question tool or if a different IRC estimation approach could be used. This question was raised as a possible approach for reducing test time, and the research team felt that they could support consistent manual entry of IRC values if additional, minimalist help screens were provided (instead of the estimation instructions typically provided, see Figure A.7). Since we already knew that manual IRC estimation, when permitted as any decimal value between 0 and 1, probably would not be consistent to more than $\pm .2$, we decided that reducing the choices to three well-defined possibilities (low, medium, and high) may support quick assessment with fairly high consistency. Therefore, the team created three brief help screens, one per IRC parameter (see Figure 7.4 for an example), linked near three radio buttons that could be used to specify each parameter on the faceted search screen.

Using the modified search screen and the claims adding page (with the form-based, analytical IRCresults questions to calculate user's model IRC values), the research team tested 11 undergraduate HCI students. Each student went through four rounds, two as a claim *searcher* (as if they were searching for a claim to use in a given design problem), using the modified search screen, and two as a claim *classifier* (as if they were entering a given claim into the system), using the claims adding page. Each round used a claim from a different notification system, although the notification systems selected were thought to be familiar to general users (cell phone vibration, AOL instant messenger alerting, Norton anti-virus pop-ups, etc). The hypothesis of the study was that students would be able to classify and search for claims in a manner that would allow them to enter and reuse each other's claims, using the four classification dimensions of the claims library

(IRC values, primary task environment, notification task, and design abstraction).

Full details about the study and findings are available in [49]. However, we summarize the most important findings here:

- **The non-IRC dimensions were effective for narrowing the search results.** We observed that in all cases, at least one the three classification dimensions other than IRC was matched. In 88.5% of all searches, two or three other dimensions were matched. This match tendency should filter most of the claims, but we rely on the IRC values to prioritize and further filter potential matches.
- **IRC values alone sometimes led to claim-finds.** Half of all classification efforts allowed at least one of the searchers to find a claim based on the IRC-values alone (perfect match). Three of the eleven classifiers had multiple searchers find their claims based on IRC values alone.
- **IRC estimation of claim classifiers (using IRCresults) improved with practice.** Second-round classifiers consistently provided IRC values that were more closely match by searchers. Experts with previous experience at estimating IRC values (tested later) were able to achieve the most consistent results.
- **Even limiting manual IRC estimation to three choices, novice searcher interrater reliability was too low.** This implies that novice searchers should be strongly encouraged to use the IRCspec system to obtain design model IRC values, at least until they are able to make estimations that similar to those generated by IRCspec.

Overall, this study was encouraging for several reasons. First, undergraduate students were very interested in the potential of claims reuse and had many ideas about how to improve the claims library. We were able to create a study that practiced interface development and user testing and analysis skills–a study that was informative to the continuing development of the claims library. Second, reviewer feedback showed external interest in our general approach to classifying and retrieving reusable design knowledge. We believe that studies like these will continue to guide improvements to the claims library and make contributions to the larger HCI research community.

User study #3: Informal observations regarding claims reuse. Not all of our early observations about the potential of the claims library were encouraging. Although we started developing the claims library with the assumption that claims would provide a strong basis for design reuse and allow designers to quickly recognize important requirements-related tradeoffs, some of our experiences cause us to question that initial assumption. Certainly, claim reuse suffers from the “not–invented–here” syndrome which challenges other forms of reuse [117]. Designers are less apt to trust a component that they did not produce themselves. The following anecdotes illustrate some of the other challenges we observed for supporting claims reuse.

Perhaps the most disturbing initial indication was observed during pilot testing. A colleague within the notification systems research group had generated several of the claims within the past

3–4 months that were included in the system. One of the design problems on a user test targeted the selection of these particular claims and the associated system. The design problem used to motivate the claim selection was essentially a summary of the requirements that guided an actual ongoing development project in which our colleague was a central designer. We asked this colleague to participate in a pilot test that would verify the clarity of the test instructions, certain that he would be able to refind his claims.

- When this pilot participant was unable to find the relevant claims with the search mechanism, it was slightly disturbing. However, when the participant was pointed to the claim (which he had written several months ago), he still did not recognize it as being useful for the design problem or associated with the system of interest!

This suggests that claims may not be such an enduring record of design knowledge as we initially thought.

Another key observation occurred in the last meeting of the semester of the class project group that initiated development of the claims library. At this meeting, it was recognized that three of the group's members were also participating in a Usability Engineering course (taught by the same professor). Both of the individual assignments and the final group assignment involved designing systems using a scenario-based design approach (a final design requirement was actually for a notification system). Students progressed through each of the phases of interface development, generating claims based on their design ideas. In many cases, this involved "burning the midnight oil" with group members, pondering what would be the most unique, yet functional set of features that could be included in such systems.

- Not one of the members of this group involved in the Usability Engineering course considered for a minute using the claims library, although they were directly involved in the content creation process for the majority of the same semester.

It was not until reflecting on the past five months that any of them made the connection–after all their assignments had been completed, submitted, and graded. When asked why they had not considered using this plethora of design knowledge, created for exactly that purpose, and readily accessible to them, they could only respond with embarrassed looks and weak excuses.

- Interestingly enough, the course professor also did not consider the possibility that a group in the Usability Engineering class might or could benefit from using the notification systems claims catalog (while completing their notification system design project).

This experience further begs the question about the suitability of claims reuse. Perhaps designers prefer to generate their own claims through the scenario-based design reasoning process than to browse through or adopt existing claims. The next section presents more reflection on these findings, and Section 7.2 introduces a potential solution.

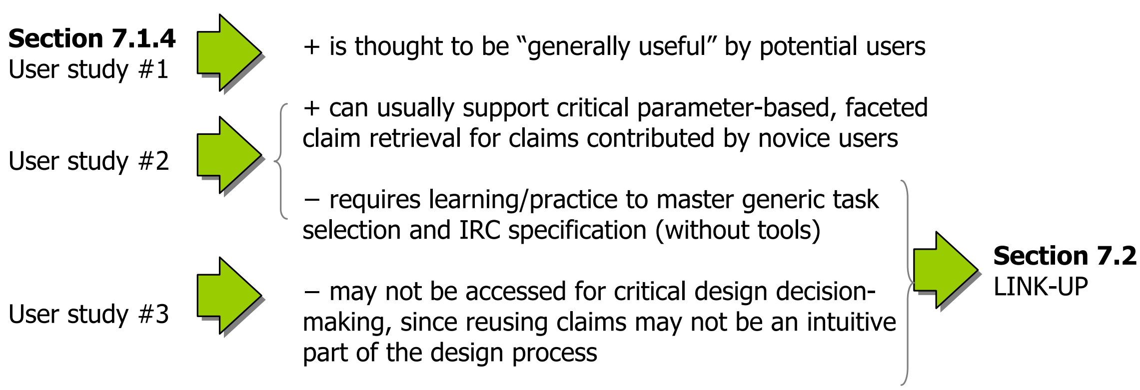
7.1.5 Discussion and future work

Based our our initial work and user testing, we are still generally encouraged by the potential for design knowledge reuse through claims. We have been able to successfully integrate the IRC framework with a notification systems claims library, as a key part of the faceted classification approach. IRCspec, the tool developed an earlier phase of research and discussed in Chapter 5, was integrated in the claims adding and editing tool, allowing claim contributors to specify accurate and consistent critical parameter values associated with the design model they had in mind as they wrote the claim.

By integrating a tool like IRCspec (one that helps a user pinpoint specific critical parameter levels for a certain scenario of use) with the claim writing activity, we may be altering the nature of claim writing itself. That is, the experience of progressing through the questions within the wizard may prompt designers to think about psychological effects related to interruption, reaction, and comprehension factors. While all of the claim upside and downside points might not relate to the critical parameters, perhaps many of them will. This possibility leads to an interesting relationship between the full claim record and the claim's IRC rating–the IRC rating being an abstract or factored version of the claim tradeoffs. If critical parameters really are critical for a system class (as they should be by definition), a database of knowledge that contrasts a wide variety of related design solutions would certainly be a valuable design resource.

Reflecting on the difficulties observed with claims-based design knowledge reuse and as we began to consider how IRCresults tools (see Chapter 6) could be integrated with the claims library beyond the claims adding activities, a broader vision for the claims library began to emerge. This vision–an integrated design environment–is the focus of the next section.

Claim 10 – The notification systems claims library...



However, we can identify many other directions worthy of future work for the core claims library, as summarized below:

- **Improvement of the claims library user interface.** The Domain Theory implementation of the claims catalog uses four key search dimensions to identify appropriate reusable content.

The primary purpose of the user interface is to facilitate selection of these terms, especially by providing users with appropriate information to make their decision. This quality of the interface can be vastly improved. Search results and individual claims also can be presented better to users. Ideally, the interface should allow full service collection of claims, such as with the shopping-cart metaphor that many e-commerce website employ.

- **Content additions.** Although the catalog has been initially populated with about 130 claims relating to about 40 systems (and 151,000+ autogenerated claims), there is much design knowledge for notification systems that should be added as this catalog matures. This effort will be greatly facilitated by the successful implementation of the maintenance features. New content can come from class design projects, related literatures, and system development efforts at other research locations.

- **Enhanced library searching mechanisms.** Digital library research has resulted in many techniques and algorithms for increasing relevancy and precision of search results. The current mechanisms driving the catalog searches are quite simplistic and can benefit greatly from such research application.

- **Inclusion of recommender functions.** As a longer-term project, the catalog can be enhanced with the inclusion of recommender functions, perhaps with operate based on an established user profile. If the system can store information about a catalog user's usage history and patterns, as well as content additions and evaluations, rule-based suggestions could facilitate cross-domain knowledge identification.

- **Additional user testing.** Use of the catalog to support design by reuse must be probed in a variety of ways. Lab-based testing can assess specific questions, such as the tendency of designers to reuse claims vs. adopt those of others (comparing results to adoptions of patterns and other forms of reusable design knowledge would also be interesting). Repeated testing on the initial user test, especially using think-aloud protocols and close user observation, would be useful. However, the most interesting results will probably come from evaluation of the catalog in context, as it actually supports notification design during long-term projects.

- **Exploration of other reuse paradigms.** Other reuse paradigms, especially patterns, certainly should not be ruled out and can serve as an interesting comparison basis for this Domain Theory implementation.

It is hoped that our initial effort can provide a start to many future researchers and will lead to a deeper understanding of design reuse. The next section begins with a general assessment of the status quo achieved by this project, and then develops a broader vision for the use of the claims library.

7.2 Using the IRC Framework in an Integrated Design Process

Throughout various parts of this research report, we have speculated about the efficiencies that the IRC framework can lend to design cycle processes, especially those dependent on system description, design/redesign creation, and interface evaluation. In this section, we review the progress we have made thus far in achieving these objectives. Since an essential goal of this research program is to provide a foundation that will allow many valuable directions for future work, we lay a vital portion of the groundwork necessary for integrating the IRC framework in an integrated design process. Again, the intent here is to present the exploratory work toward developing research infrastructure to broaden the impact of the IRC framework and associated tools, opening new questions and opportunities that will guide several years of continued research.

7.2.1 Assessing the status quo

Most of the research described in this document was designed to address the reviewer comments listed in Section 4.5, particular those that found limitations of the IRC framework based on subjective rating assessments, lack of empirically demonstrated and tested functionality, or the utility available beyond a human information processing description. We have informally speculated about many other ways that the IRC framework can be useful, as depicted in our initial vision of the IRC framework in Figure 7.5. We have built tools to support consistent and accurate IRC classification performance and developed a notification systems claims library that uses the IRC as a primary index. However, we have also recognized that more is needed before the IRC can truly impact a notification system design process. Developing a vision to fill this need is an essential first step toward creating new hypotheses for the role of the IRC framework in notification systems research, HCI education, and interface design processes.

This section introduces a vision for an *integrated design environment* for notification systems, which should mitigate the issues uncovered in the claims library user testing and observation described in Section 7.1.4. That is, we are proposing a software environment in which designers are able to proceed through typical early phases of a system development cycle, supported by tools, in which they access and contribute to the claims library. The centrality of critical parameters as a systematic gauge of iterative and long-term progress remains prominent in this vision. Our general thought is that *when valuable design tools embody critical parameters and are coupled with readily accessible reusable design knowledge, interface development will improve as a scientific endeavor.* Arguments originally introduced by Carroll [26, 29, 108, 23, 27, 24] and Sutcliffe [116, 117, 118] can be reified through this approach.

Revisiting the observations regarding claims reuse described earlier, a few key points and implications should guide development of a integrated design environment:

- **Potential users saw more value in the claims library for inspiring new ideas than supporting existing design ideas**–ideally, users should value both activities. This could imply that claims were not perceived to be grounded by evidence that was stronger than their ex-

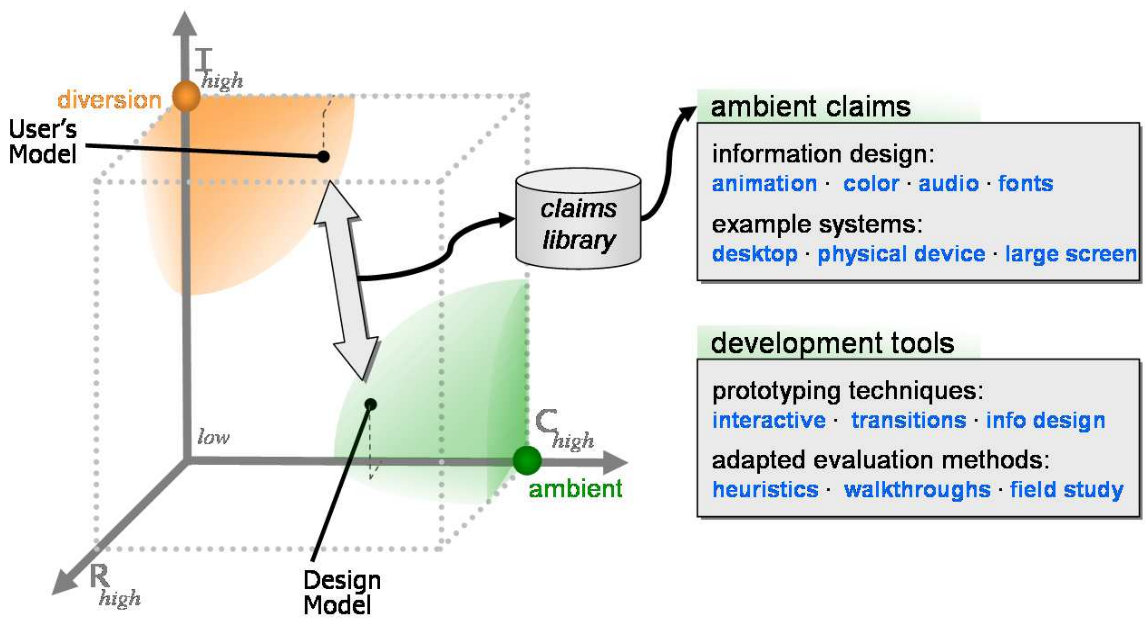


Figure 7.5: Design model and user's model plots of a notification system, revealing the reengineering approach necessary. Reengineering approach trajectories may be reusable, revealing common design problems. Design models can be associated with development resources that are specifically adapted for the combination of user goals. In this example, the design model is an ambient interface (a notification system sub-class, characterized by IRC region), so a design catalog would return claims and other resources that have been shown to be successful for that blend of user goals.

isting ideas, and/or that these novice designers generally do not recognize a need to support design ideas at all. An integrated design environment may be able to address both implications, even though the later implication may be typical within the interface design culture.

- While retrieval mechanisms were usually effective for returning potentially reusable claims for a given design problem, **not all potential users were successful in finding claims**. This implies that claim finding and reuse might not be possible for all users unless robust, context-sensitive support is provided by a retrieval system. An integrated design environment could provide the context necessary (characterized by the design stage in which the retrieval features are accessed, as well as information related to earlier design rationale) to enable recommender features.

- **Potential users were observed missing key opportunities for claims reuse in real design activities, although they were quite aware of the claims library content.** This implies that claims reuse, and claim contributions, may not happen if access to the claims library is not an intuitive, natural, low-overhead activity closely connected to other design activities. With an integrated design environment that offers support for typical design processes (i.e., problem definition and task modeling, usability evaluations, and system documentation), a situation may finally be created where reusing an existing claim (and design artifact) might be easier and more appealing that recreating one from scratch.

The broad vision for this system provided here (Section 7.2.2) has appeared in the 2004 Conference on Computer-Aided Design of User Interfaces (CADUI) [32], and has been funded for implementation. As we develop the task-related features and conceptual basis of the integrated design environment, we recognize that other actions must be taken to promote and ensure use by notification system designers. In the next chapter, we describe two aspects of the promotion strategy: integrating the design environment with HCI learning process (see Section 8.3.1) and using the system to facilitate distributed, multidisciplinary development efforts and design research (see Section 8.3.2).

This portion of the project was primarily undertaken to develop ideas and opportunities for future work and extend the potential of the IRC framework, and much of this material is still formative. As emphasized at several points in this document, we regard the design area of notification systems as merely an example design area to probe ideas related to usability engineering and design knowledge reuse. We speculate that many other areas of interface design can be framed with their own critical parameters and augmented with an integrated design environment similar to the one we propose.

7.2.2 A broader vision: LINK-UP

To support design knowledge reuse and growth for notification systems, several arguments from the Computer-Aided Design of User Interfaces (CADUI) research community are influential. Since notification systems design is inherently focused on supporting primary and secondary task performance, approaches that seek to understand and model desired task behavior are key. In particular, the Enhanced Task-Action Grammar (ETAG) provides a proven mechanism to describe interface expectations and deliberately connects HCI and software engineering concerns [43]. Wilson and Johnson present considerations for task-based models that would assist in developing the connection between design phases, identification of optional and compulsory features of the existing task model, and development of the envisioned task model [136]. This general approach seems promising, as we consider how a practice of claims reuse might be better integrated with other design activities.

Building on this foundation, we propose an interface design process compatible with scenario-based design methods and our claims library (see Section 7.1), but specifically intended to facilitate three primary goals: design knowledge reuse, comparison of design products, and long-term research growth within HCI. For example, a designer of a notification system for collaborative work status should be able to benefit from lessons learned in developing previous, similar systems–perhaps a notification system for news headlines or weather information. Claims about appropriate artifacts used in other domains can be accessed for reuse by designers to meet user notification goals. In conceptualizing and developing this system, we have determined that critical parameters provide a meaningful mechanism to specify and describe claims, allowing structured design process transition and reuse.

LINK-UP, Our envisioned system. The *LINK-UP system* (Leveraging Integrated Notification Knowledge through Usability Parameters) operationalizes our proposed interface design process. The root concept of the system is to provide notification systems designers with a facility for task-based design advice, consistent with the Wilson and Johnson definition [136], guiding progression throughout an interface design process. This design advice comes in the form of claims from the associated claims library. Claims stored a design knowledge repository are accessed and modified at several points within the interface design cycle using interactive system tools. Figure 7.6 depicts the general architecture of the LINK-UP system, which is composed of five inter-related system/activities–*requirements analysis* (#1 in the figure), *participatory design negotiation* (#3), *analytical evaluation* (#4), *empirical lab-based testing* (#5), and the core *claims library* (#2). The four design-support modules (which include all systems/activities other than the claims library) are described in sections that follow. We also discuss a concept for relating claims that has not yet been formalized as a software tool.

The LINK-UP system is intended to support a scenario-based design process [23, 24, 108] and approach to claims reuse [29, 116, 118], with its use of consistent terminology and underlying design philosophy. However, in some instances we have had to extend basic terminology to support new activities that we propose. For instance, while *claims* have been defined by Carroll et al. [28], and the Rosson and Carroll textbook *Usability Engineering* describes four types of scenarios that claims can elaborate (problem, activity, information, and interaction), we refer to two specific types of claims:

- **Problem claims** – articulate key points of a problem scenario, expressing situational con-

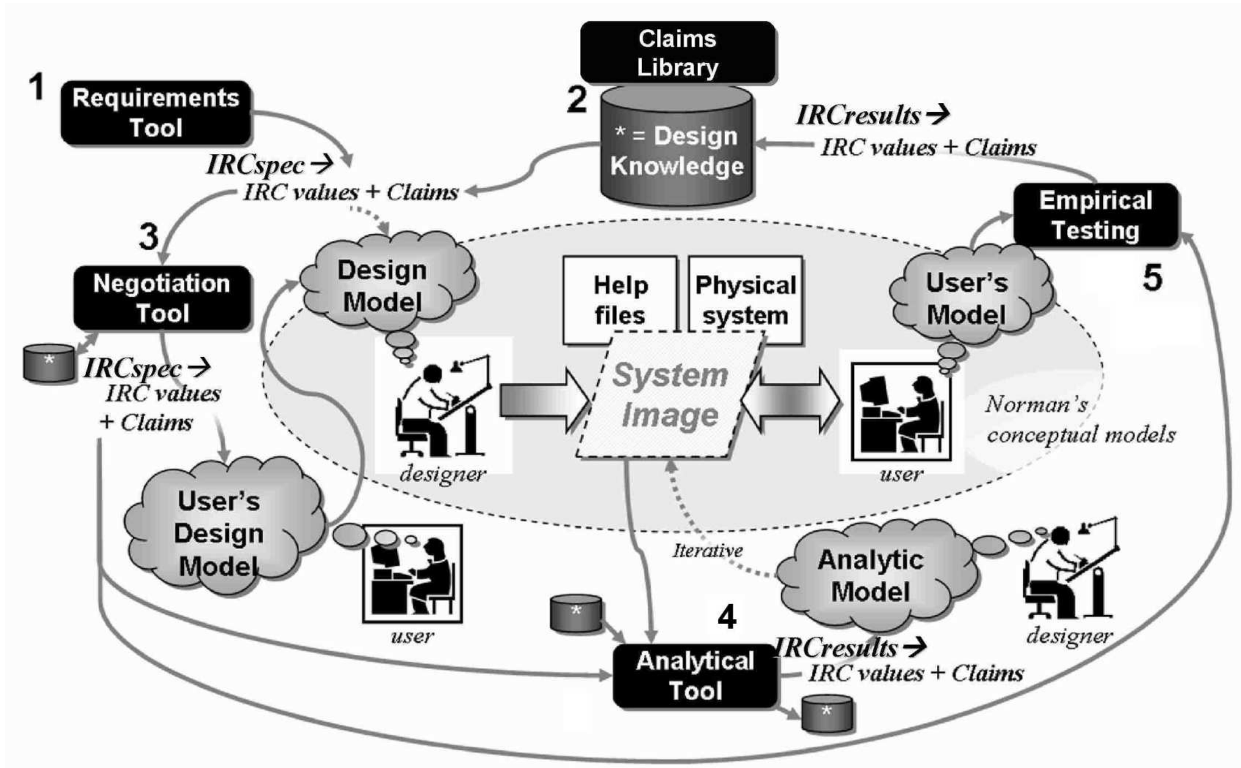


Figure 7.6: General architecture of LINK-UP's design-support modules. The light grey region in the center depicts Norman's conceptual models (Norman, 1986), which are extended through our work. Numbers refer to steps though the process, and are referenced and explained in the sections that follow.

straints, observations about the targeted user's background, and other requirements-level tradeoffs that might later contribute to psychological consequences in a typical usage experience. Using the IRC framework, a design model IRC would summarize a set of problem claims.

- **Design claims** – describe the psychological tradeoffs expected or actually resulting from design decisions expressed in activity, information, or interaction scenarios. Design claims provide more details about the user's model IRC.

These two terms are used throughout the description of LINK-UP activities and help expose the benefits of characterizing a design model and user's model with critical parameters.

When designers use LINK-UP, they are generally performing activities within a web-based project object associated with the interface they are designing or evaluating and their user account. The progression through the modules is intended to be loosely linear, although iteration within a given module(s) and skipping between modules is also expected. While we envision that LINK-UP will also include other modules that facilitate learning and research, we focus here on articulating the high-level vision of the design-support modules. The initial vision for each of these modules was originally developed with a small group of researchers and published in [32]. However, further development under our supervision has been done by 22 students in a graduate class[9] and an undergraduate seminar [10].

7.2.3 Module #1, Requirements Analysis

General vision. This series of steps within the requirements analysis module starts with the problem scenario and results in a *template* for connecting problem claims by stage of action (i.e. Norman's stages of action [100]. The tasks, information characteristics, user background, and other aspects of the situation from requirements gathering in this step, combined with previous design knowledge, formulate the design model. Within the module, various processes assist the designer, such as selection of basic tasks (generic and generalized), hierarchical task analysis, matching of specific requirements to standard (general) task models, and decomposing a task model to stages of action (see Figure 7.7). Some of these processes can be accomplished by simple forms while others require fairly robust tools or a more complicated series of interactions.

A key objective of the module, and a step toward creating the template for problem claim selection, is the decomposition of tasks by stage of action in order to specify one or more typical action sequence (as a task model) that would be carried out through the use of the designed interface. That is, the designer might decompose a "Respond to Important Alarm" meta-task, which is an essential user interaction with the notification system expressed in the problem scenario(s). When they have completed this process, they will have identified the generic tasks that must be accomplished to proceed through each stage of action. For example, in the first stage of the gulf of evaluation, *Perceive*, one generic task might be *identify*. Figure 7.8 depicts this concept.

Since the design module IRC for the system has already been established at this point (after creation of the problem scenario(s)), the designer can also think about how the overall IRC for the action sequence changes as the user moves through each stage of action. For instance, while the Perceive stage might require a low level of interruption, a drop in primary task sustainment (a component variable of interruption) might be acceptable in later gulf of evaluation stages, so interruption would rise in the *Interpret* and *Making Sense* stages.

LINK-UP assists designers with both decomposition of the meta-task and design model IRC for the action sequence. Available as a starting point for their analysis is a collection of task models (like the one shown in Figure 7.8) for each of the IRC scenario families (i.e. corners of the IRC cube, described in Section 4.2.2 and Table 4.3). Based on the design model IRC estimated with the IRCspec tool, specific task models are recommended. Since the pre-existing task models include IRC trends within the stages of action as well as composite generic tasks, they serve as a complete template for problem claim selection (which can be modified by designers as needed). With this template, a designers is provided with two of the primary indices for claims searching–the IRC

[9] Spring 2004, CS 6724 Design and Software Reuse for HCI; acknowledgement of individual student contributions provided in appropriate context

[10] Spring 20004, LINK-UP seminar through the VTURCS program (Virginia Tech Undergraduate Research in Computer Science)

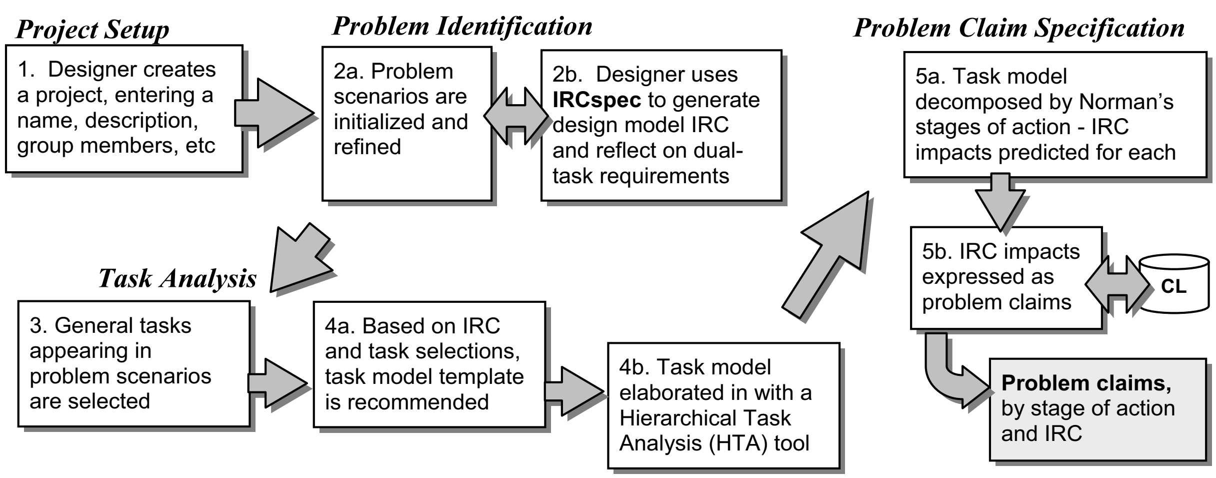


Figure 7.7: The high-level activities supported by LINK-UP's Requirments Analysis Module (external HTA tool is not currently integrated.

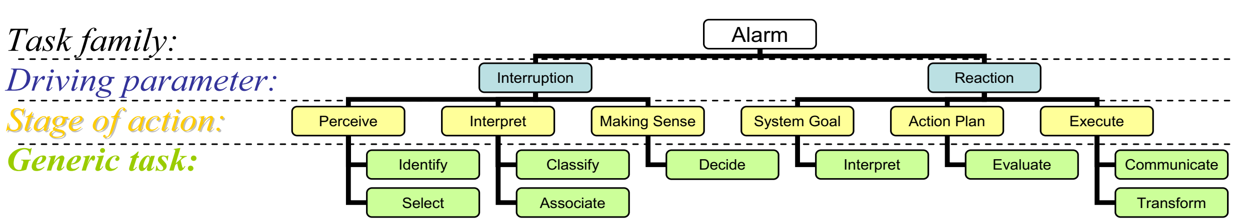


Figure 7.8: An example of a stage of action task decomposition for an Alarm task, such as "Respond to Important Alarm."

values and the notification task. Filling the template with problem claims from the claims library or their own creation ensure task coverage (as defined by Carroll et al. in [29]).

IRC integration. Interwoven throughout the process of completing a requirements analysis with this module is the systematic questioning of the IRC critical parameters. A designer begins this questioning after they have initially recorded a problem scenario and they are prompted to specify a design model IRC with IRCspec. Since this tool helps designers think about facets of the dual-task situation, it is likely that a novice designer will revise their problem scenario after using IRCspec. The overall design model IRC, along with the basic tasks identified by the designer, is used to form a recommendation for a base task model and problem claim template.

A more detailed questioning the IRC parameters and development of the design model occurs once the generic tasks are analyzed in the emerging problem claim template. Here, the concrete terms that compose the IRC parameters (as discussed in Section 6.3.1), as well as the notification action models (see Section 4.2.3), provide some structure for reasoning about parameter changes that should occur as the user moves through the stages of action. Thinking in terms of broad trends still allows recognition of subtle changes in the overall parameters.

Current status and ongoing work. A team of developers[11] have implemented a prototype of this general vision for the requirements analysis module. The prototype currently includes four pre-existing task models and stage of action IRC breakdowns, although it is expected that at least a few models for each of the eight IRC families would need to be added. Initial user testing has been completed, validating that the general steps within the module are comprehensible by undergraduate HCI students.

Plans have been made to integrate a third-party open source tool that allows construction of a hierarchial task analysis. Using the pre-existing task models, a designer would be able to use task-related information to express the specific requirements of their design problem. In upcoming months, broader activities like this will be developed, and usability improvements will be made to the existing portions of the current implementation. It is anticipated that a fully functional version of the module will be ready for pilot testing in a Spring 2005 undergraduate HCI class.

7.2.4 Module #3, Participatory Design Negotiation

The third module can be used at several different points: as an alternate starting point for the design process, as a follow-on step to the requirements analysis module, or as a process that can be returned to later after initial evaluation results have been obtained. However, we describe the general vision for the module with the assumption that the designer has first completed the tasks within the requirements module that would result in specification of problem scenarios and claims.

[11]Dillon Bussert, Cyril Montabert, Solomon Gifford, and Melissa Grant

General vision. The participatory design negotiation module is intended to guide planning, execution, and analysis of an interactive design discussion with stakeholder users. *Participatory design* is an approach to interface development that involves potential end users in decision-making throughout formative stages of the development cycle. When designers sit down with users in a participatory design session, there are many possible negotiation points. Our thought is that a module in LINK-UP can help a designer prepare for a meeting with stakeholders, ensure that some discussion focuses on issues related to critical parameters, and archive meeting results that might be useful to other designers with similar requirements. As with other modules, claims serve as a primary unit for design deliberation. Therefore, this module integrates a participatory design negotiation technique with the claim development and reuse process in the LINK-UP system.

Starting with problem claims in the stage of action templates that emerges from the requirements analysis module, this module helps designers present understanding of requirements to stakeholders and receive specific feedback. Tools within this module allow a designer to build a participatory negotiation session and allow a stakeholder to take part in one. Although the actual session is intended to be a face-to-face focus group format that includes a few brief transactions with the web-based LINK-UP system, the web-based nature of the system would also support distributed meetings.

IRC integration. A key theme throughout the subprocesses within this module is the receiving of stakeholder feedback on design model IRC levels according to individual claims. Problem claims describe aspects of the current usage situation, which communicate instances where user notification goals are needed. These goals are likely to have implications for user interruption, reaction, and comprehension, both in the form of claim upsides and downsides. The explicit decision that designers make when expressing critical parameter impacts as either an upside or downside directly contributes to the design model IRC estimation. Ensuring that these expectations are valid, in the context of a reasonable problem scenario and according to stakeholders, is the essence of LINK-UP's participatory negotiation module.

Therefore, designers are assisted with including questions and considerations that will generate participatory comments related to claim IRC levels. For example, designers will discuss a claim about a situation feature and probe stakeholders on whether they agree that it suggests a need for a moderately interruptive notification, or whether they feel that only a small portion of all information available would be important to display. In this manner, designers can make inferences about appropriateness of specific tradeoffs expressed by claims as well as the design model IRC(s).

As the design model IRC is adjusted by stakeholder feedback, the *user's design model* is formed. The claims library can be searched for claims that might be more appropriate in the stakeholder's point of view. A plausible heuristic for the amount of participatory negotiation that must be done could result from an analysis of how closely the design model IRC and the user's design model generally match, established from a representative sample of potential users. If user's design model is formed from consistently expressed answers that suggest a specific portion of the IRC framework, and this disagrees considerably from the design model, designers must question the validity of their own understanding of user notification requirements.

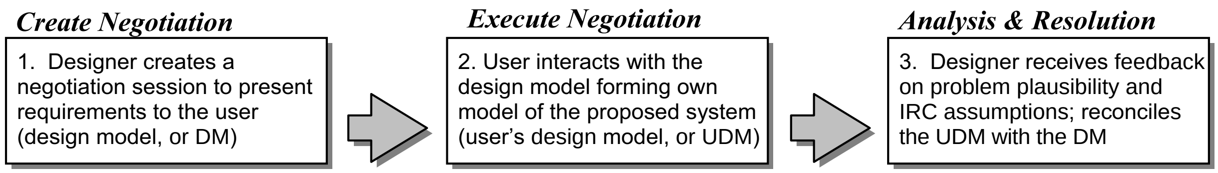


Figure 7.9: The three high-level activities supported in the current implementation of LINK-UP's Participatory Negotiation Module.

Current status and ongoing work. As a class project in Spring 2004, a team of students[12] has extended the general vision for this module, created prototype materials, and run an initial pilot test using the module. This team has developed and tested screens that allow designers to setup a participatory negotiation session where stakeholders enter feedback related to problem claim IRC values. The screens and database integration support three high-level activities, as depicted in Figure 7.9: creation of a negotiation session (by a designer), execution of a negotiation session (by a stakeholder representative), and analysis and resolution of the results (by a designer). Use of these screens and the general process of a participatory negotiation session has been tested with two designers, and the results have been submitted for publication [94].

Initial findings are encouraging: when a participatory focus group is conducted (and supported with web-based tools) with the intention to elicit feedback about appropriateness of problem claim IRC levels, both designers agreed that the feedback obtained was useful and would guide reconsideration of basic requirements. Continued testing may convincingly show that the general strategy of questioning expected critical parameter values as the basis of a group discussion may prove to be an effective way to involve stakeholders in design decisions, without dwelling on irrelevant details. Archiving results of negotiation sessions (where a claim's IRC values are judged by stakeholders to be "too high" or "too low") offers exciting long-term prospects as this design knowledge accumulates. Many directions for future work on the participatory negotiation process and module have been recognized by this development team and others, further evidence that embodying an idea like the IRC framework in a system like LINK-UP opens new doors for HCI and notification systems research.

7.2.5 Module #4, Analytical Evaluation

General vision. Module #4 is intended to be used after a designer develops a system image (a visible or otherwise perceptible instantiation of the user interface), either through the full implementation of a working system or a minimally functional prototype. This tool helps designers create a description of the key interface features with design claims, for the express purpose of receiving feedback from an analytic evaluation. *Design claims* describe psychological effects embodied in designed artifacts (as opposed to situational factors) that result from activity, information, or interaction design decisions with a scenario-based development process [108]. Like the Partic-

[12] Ali Ndiwalana, Ian McEwan, Nithiwat Kampanya, and Kevin Pious

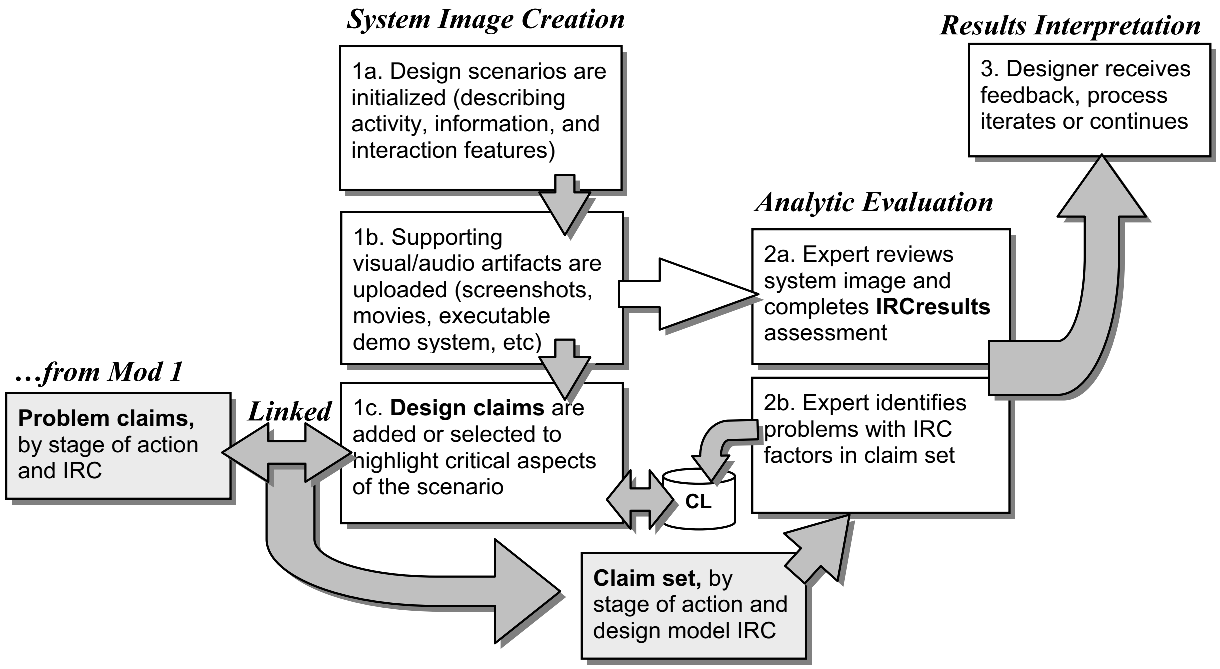


Figure 7.10: The high-level activities composing LINK-UP's Analytic Evaluation Module.

ipatory Negotiation Module, three general activities need to be supported: creation of a system image description, execution of an analytic evaluation, and presentation of evaluation results back to the designer. Figure 7.10 depicts the key processes within this module.

Although a designer may be motivated to complete the first process (creating the system image description) in order to prepare materials for the analytic evaluation, this portion of the module serves more broadly as a mechanism for building design knowledge stored in the claims library. Rather than rely on contributions that occur outside of valued design activities, mechanisms like this should ensure that LINK-UP's claims library will effortlessly grow as a by-product of use (similar mechanisms are build into all modules). Revisiting the stage of action template that holds the problem claims (established in the Requirements Analysis Module, the designer links each problem claim to a design scenario(s) and design claim(s). Design claims can be reused completely or in part from claims within the repository, or entered as original entries. Whether new or reused, a key process is the association of artifact representations (screen shots, pictures, etc) with claims. This process of specifying the design claims and representing prototype artifacts expresses the system image.

IRC integration. When the system image is established, designers are able to revise design model IRCs for problem claims. However, with the completion of the process that links problem claims to design claims, a designer is explicitly asserting that a design claim embodies a given design model IRC.

In the analytic evaluation, an expert evaluator familiarizes themselves with the system image, and then uses the IRCresults questions to establish a general user's model IRC for the system (more precisely referred to as the *analytic model*). From here, an expert considers each "claim pair" (i.e., a problem claim and the corresponding design claim) to determine whether the artifact supports the desired IRC or not (similar to the participatory negotiation process). Like the process of questioning IRC trends through the stages of action for problem claims (occurring in the Requirements Analysis Module), evaluators can reason about the IRC effects caused by design artifacts with the concrete IRC variables (see Chapter 6). For instance, an evaluator might recognize that a particular artifact (or artifact state in a stage of action) would cause a sharp decrease in a user's primary task sustainment, thus increasing the I-value. Comments are provided to elaborate on expected differences between the design model and analytic model IRC for each claim.

Evaluator feedback is stored as part of the designer's project, but is also appended to a claim record. Therefore, when a claim is considered later by another designer as a potentially reusable component, the original designer's intentions as well as the evaluator's assessment (and characteristics of user performance, as we describe in the discussion of Module #5) is available to assist decision-making.

The notion of claim-level and system-level IRC values presents some interesting considerations. While it was originally intended that IRC values be used to characterize an artifact or situation of any granularity (i.e., an entire system situated in a context of use or a specific design feature within a system), activities within this module reveal tradeoffs related to assigning IRCs. A viable alternative to specifying a precise numeric parameter values for artifacts of any granularity is, of course:

- Situation or system-level IRCs should/continue to be determined by the IRCspec and IRCresults tools,
- Claims should be described in terms of the concrete effects (IRC factors) it may generally have (e.g., "sharp decrease in sustainment" or "no effect on projection") and the impact it may have within the complete task flow through the stages of action (e.g., "very brief or minor effort" or "will dominate the usage experience for several seconds"),
- Claim-level IRC factors can then be used within the stage of action framework to identify where or how reusable design components deviate from or contribute to the intended task coverage expressed by the design rationale,
- Claims should be characterized and retrieved by dominate IRC factors.

This approach has been made possible by the research presented in Chapter 6 and is thought to be a promising direction for the continued development of the claims library and LINK-UP system.

Current status and ongoing work. A prototype analytic module has also been implemented and tested by a team of developers[13]. This prototype included full support for the creation of the system image description, execution of an analytic test (integrating the analytic version of IRCresults), and interpretation of the evaluation results. Each of these processes was tested in a separate pilot test session, consisting of seven different notification system designers. When the pilot test sessions commenced, the seven designers had spent about two months developing their own version of a notification system. All designers had received numerous hours of instruction on the IRC framework and claims analysis in seminar meetings, and had completed an Introduction to HCI course using the Rosson & Carroll textbook ([108]). In the pilot tests, each designer used the module to archive their system image and create claims sets of linked problem and design claims. Then, each system image was evaluated by one of the other designers (that is, designers acted as evaluators for a different system) and results were interpreted by the original designer.

During all three sessions, the developers observed how well the module supported the intended interactions and collected feedback on the usefulness of the general approach. Although many notes were made that addressed particular design decisions within the prototype module, it was very encouraging that the approach was thought to be useful to the designers, who agreed that the evaluators comments could guide redesign efforts ("*I was able to identify design problems or places where design claims should be changed*" and "*the evaluation results could be helpful in subsequent stages of the design process*") and "*the general steps in evaluating the system image make sense*." While not all designers agreed with the evaluator's analytic model IRC assessment (only one evaluator analyzed each design in this pilot test, although the module is intended to be used to collect opinions from multiple evaluators), all designers indicated agreement or strong agreement that "*the overall evaluation comments and estimated IRC value were useful*." The results of this testing (as well as details about the Analytic Module) have been submitted for publication [75]. This effort has shown much promise for the conceptual vision of the LINK-UP system and integration of the IRC framework within a design process.

7.2.6 Module #5, Empirical Lab-Based Testing

General vision. Basically, the Empirical Testing Module will facilitate creation, execution, and interpretation of lab-based, user's model IRC test results, obtained on a standardized evaluation platform. In Section 6.5, we described an automated test platform that was used to capture user performance data necessary to infer user's model IRC values for three different notification systems. As a user played a block catching game (simulating a primary task), they monitored the notification system under evaluation and reacted to data changes. While we were satisfied with the general evaluation approach and recognized that it could provide a platform for accumulating data about critical parameters for reference tasks, we also realized that it would still be a complicated test for others to replicate, limiting the impact of the empirical version of IRCresults. That is, with a data set like the one included in Appendix B.12, B.13, and B.14, a designer should be able easily change the test conditions (e.g., the signal criteria and timing definitions, the preset cost of inter-

[13] Jason Chong Lee, Sirong Lin, Alan Fabian, and Andrew Jackson

ruption, questions to measure base comprehension and projection, and the notification system used in the testing) and use this testing platform. An Empirical Testing Module within LINK-UP could help novice designers perform replicable, empirical testing with several reusable test platforms.

While the high-level activities in this module also follow the pattern of "create a test, test, interpret a test" (like Participatory Negotiation and Analytic Modules), there are many unique considerations for this module, especially in the "create a test" process. As designers proceed through the design process supported by LINK-UP, they generate a large collection of claims. Individual claims that have been used in other design efforts or are based on research might include empirical supporting evidence, but often they do not, and rarely are all sets of associated claims in the interface adequately validated. Low-cost feedback obtained from expert evaluators using the Analytic Module, or potential user stakeholder in a Participatory Negotiation, may be enough to validate most design features or convincingly support many reusable claims. However, it is expected that designers will still want empirically validate a few essential claims–making this process easy and replicable will add value to the claims library, enabling the creation of comparable, empirically grounded design knowledge.

An initial key process within the Empirical Testing Module is assisting the designer with the process of selecting essential claims to test. This process revisits the problem–design claim set(s) established during creation of the system image in Module #4, perhaps focusing the designer on claim pairs that still exhibit a wide difference between design model IRCs and user's design model or analytic model IRCs–the designer might wish to establish through empirical testing that the claims will produce the predicted psychological effects (design model IRC matches the user's model IRC), and that the sample users are generally satisfied with the notification system.

After the designer selects key claims to test, and indicates time intervals during which the claims are expected to be "active," the system should assist the designer with identifying and setting appropriate test conditions, such as those identified in Figures B.13, and B.14. Multiple test platforms would be available for the designer to select from (with new platforms being added into the system as desired), so that the most ecologically valid testing experience (to include primary task simulation) could be supported. As a result of the interaction, a test script would be generated for the designer, which would drive the reusable, automated testing platform (available for download or executing as a web-service). Since the test platform would generate a output file for each participant, the designer would simply gather all files and upload them back into the Empirical Testing Module for analysis. Of course, claim records of applicable claims would be appended with testing results.

IRC integration. Since the testing platform has been designed to capture user performance-related metrics that are required for calculation of the IRC parameters, this module is closely integrated with the IRC framework. Perhaps the most interesting new idea here is the segmentation of an actual (or simulated) usage experience so that data related to "active claims" may be captured. To clarify, a designer may have a claim that describes the effects of an audible chime that occurs when certain notification criteria are met (i.e., a price changes, or a new message arrives). Using a pre-defined data set, a designer is able to determine when the chimes will sound (i.e., at times =

:37, 2:12, 4:32, and 7:13). If the designer is interested in observing the validity of the claim, they might say that it is “active” for about ten seconds after each chime (i.e. for times :37 - :47, 2:12 - 2:22, etc.). During those time periods, user performance metrics will be used to infer claim IRC factors.

In a sense, this segmentation of the usage experience should generally follow iterations of the stages of action. Therefore, it is expected that the claim pairs (also organized by stage of action) will easily map to interface events. Although the entire usage experience can be averaged (weighted appropriately) to determine a system-level user’s model IRC, individual claims can also be appended with information the specified how the IRC factors changed. This approach is compatible with the approach recommended in the previous subsection (see bulleted points).

Current status and ongoing work. Like the other modules, this module has also been prototyped and pilot tested by a team of developers[14] It was encouraging that the processes embodied in this module could, in fact, be automated and used by designers. Although the current prototype requires quite a bit of further development, pilot testing showed that the module could be used by both experienced notification system designers and true novice designers. Pilot tests were performed on the same designers that tested Module #4, just after the they used the Analytical Module. This team asked the designers to create a test script, execute an empirical user test with the reusable test platform, and upload and interpret the results.

The pilot testing revealed several limitations of the current implementation–particularly the need to improve understandability and facilitation of the test script creation process. However, process was generally found to be achievable and useful (see Figure C.20 for specific questions and responses). Features that assist with the segmentation of the user experience and select of claims were perhaps most deficient. A second test conducted by the development team involved students that were not knowledgeable about HCI, claims, the IRC framework, or notification systems. These students were asked to interpret the results of the empirical testing after they had been uploaded back into the Empirical Testing Module and appended to appropriate claims. When asked to select claims that matched a given design model, most students chose claims that reflected appropriate user’s model IRC evidence–a finding supports the high-level vision of our approach to claims reuse, LINK-UP, and this module.

7.2.7 A Sixth Module? – Exploring Claim Relationships

General vision. Responding to some of the difficulties observed in Section 7.1.4 related to user formulation of search criteria to access claim library contents, we are exploring other strategies. Perhaps one of the more promising strategies leverages relationships between existing claims, suggesting a graphical visualization tool that helps users move between related claims. In both Carroll and Sutcliffe’s early work, claims are intended to be formed from factoring and generalization [29, 116, 117], as well as semantic question-asking heuristics, such as the stages of action [28] or

[14] John Booker, Laurian Hobby, Jason Zietz, and Anderson Ray Tarpley III.

"what" and "how" questions [27]. Based on these ideas, we have defined and provided examples of six formal relationship-types to describe claim relationships (in a paper published in the 2004 International Conference on Software Engineering and Knowledge Engineering, [129]):

- **Postulating/predicating** – *Postulating claim* relationship starts from evidence gathered in early design process (claim A), asserting psychological effects that are observed later as the design develops (claim B); a *predicating claim* relationship starts in a later design stage (claim A) to explain (or revise and explanation of) phenomena observed in an earlier stage (claim B). This relationship is especially useful for describing the relationship between problem and design claims (see Figure 7.11a).

- **Executing/evaluating** – To describe claim connections across Norman's stages of action [100] and Carroll's concept of task coverage [29], we can refer to claim relationships as *executing* two claims (if the second claim in the sequence addresses tasks within the Gulf of Execution stages), or evaluating (if the second claim address the Gulf of Evaluation); see Figure 7.11b.

- **Generalizing/specifying** – As discussed by Sutcliffe and Carroll [116], claims that are more abstract or factored versions of concrete claims can be said to be *specifying*, while the term "*generalizing*" can describe the relationship from a concrete claim to a more abstract claim; (see Figure 7.11c).

- **Translating** – Two claims that express the same general psychological effects, but in two different domains of use, can be said to share a *translating* relationship; (see Figure 7.11c).

- **Fusing/Diffusing** – After two claims are combined (perhaps from different domains) to form a new claim, the original claims can be described as "*fusing*" the new claim, while the *diffusing* relationship describes extraction of claims from a single claim. While "postulating" describes the movement from a problem to a solution, "fusing" describes the joining of multiple problem or multiple solution components; (see Figure 7.11d).

- **Mitigating** – A *mitigating* relationship exists between two or more claims when the initial claim expresses downside tradeoffs that are remedied with a new or redesigned artifact (described in a second claim); (see Figure 7.11d). While predicating and mitigating relationships may appear similar, the key difference is that mitigating relationships connect versions of artifacts developed in similar design processes (i.e. information design), while predicating claims transcend design stages.

Thus, our vision for a new module helps users visualize relationship types to other claims in the claims library from any claim within the design-support modules.

IRC integration, current status, and ongoing work. Integration of the IRC framework in the LINK-UP system and a notification systems development process is somewhat enhanced with the

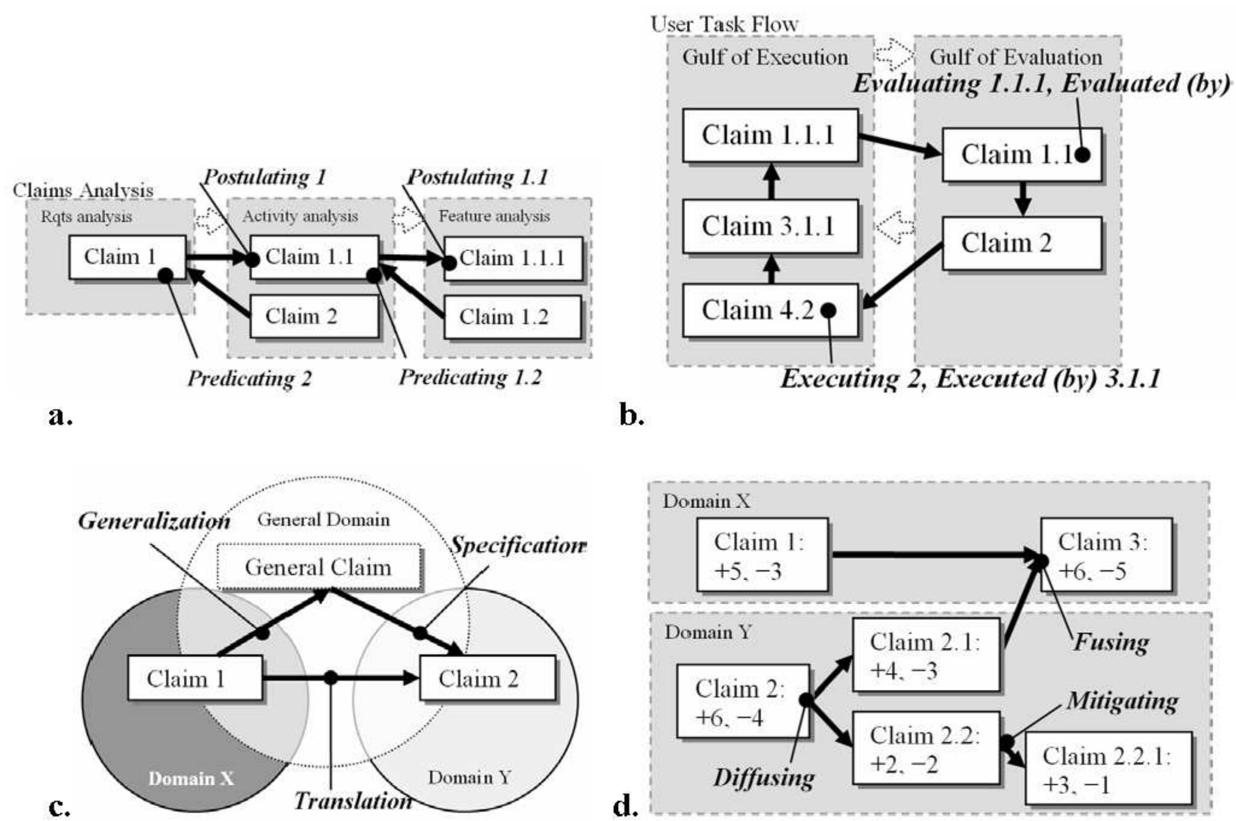


Figure 7.11: Claim relationships, described in the bulleted points above, to appear in [129].

claim relationships and relationship visualization module. Although the relationships can be based on any of the claim attributes, the IRC values or concrete factors certain can provide the basis of the relationship.

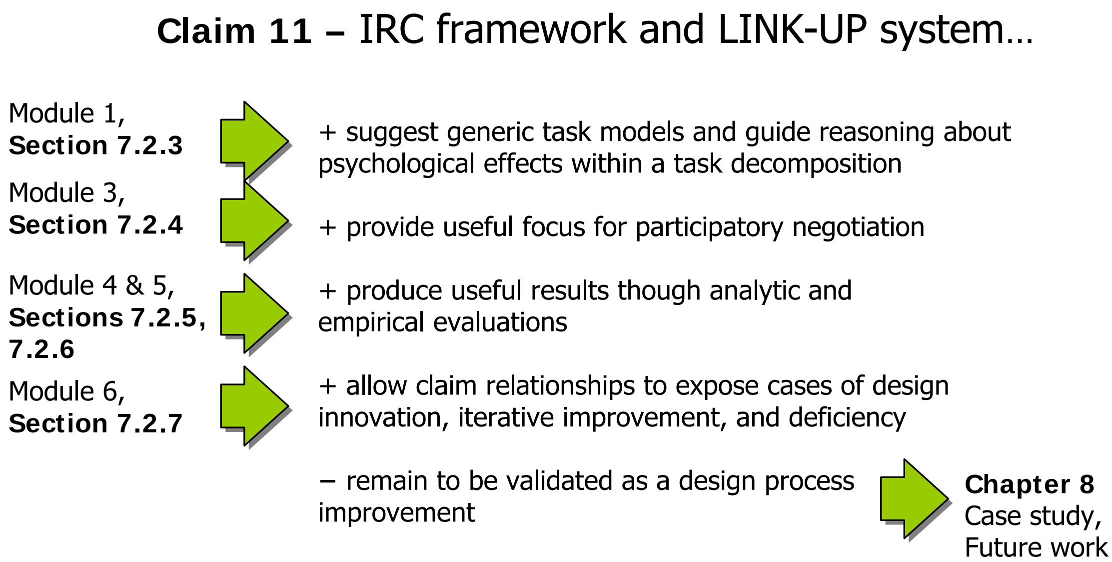
An initial version of a module for claim relationship visualization has been developed and tested by Shahtab Wahid, Jamie Smith, Brandon Berry, and Nima Rashidi. Results of pilot testing show that, when users clearly understand claims analysis and the claim relationships, the tool is thought to be a useful and powerful search tool. However, there is certainly a learning-curve involved with the current visualization and interaction technique, which future design iterations should seek to reduce. The notion of the claim relationship must be fully and seamlessly supported in claims adding and editing screens, and careful thought will be need to fully leverage these powerful relationships to recognize growth, innovation, and the maturity level of the claims library content. Explicitly recognizing claim relationships that are based on the IRC parameters will help reveal cases in which improvements are in our ability to design notification systems to meet the goals and constraint expressed by the critical parameters and the attention-utility theme.

7.3 General Discussion

Our exploratory efforts in initializing a claims library and charting an integrated design process for notification systems development have allowed recognition of several specific benefits supported by the IRC framework. To summarize, coupled with a claims library, the IRC framework serves to scope a user's task context, helping designers judge claims are relevant and potentially reusable. The LINK-UP system provides a web-based interface to guide the usability engineering process for a notification system. Designers interact with five major design support tools (including support for requirements analysis and negotiation, analytical and empirical testing, and design knowledge access), saving and building on progressive session results throughout the process. A set of claims (serving as design hypotheses) and associated critical parameters (serving as engineering targets and results) guide design progress, within a single design and through a meta-analysis of several systems. The design knowledge repository will grow and improve through use, becoming a living record of notification systems research.

In all pilot testing sessions with the LINK-UP module prototypes, we were generally encouraged that participants found the system interactions to be understandable and useful–a marked improvement over the testing with just the claims library. We found the responses from the seven notification system designers quite encouraging, as well as their continued interest in the development of the LINK-UP system. From their survey results that compared the experiences with the Analytic Module to the Empirical Testing Module (see Figure C.21), we can note that different designers seem to value the activities supported by both modules and would use them again in another design process.

As 21 other developers were involved in the prototyping of LINK-UP modules, integration of the IRC framework into each module did not happen by chance or as the result of an single opinion. Rather, the framework and its tools were implicitly accepted as a means for supporting



key processes within the modules. Several of these students have elected to continue graduate research on topics that extend or support the IRC framework and use of critical parameters in a design process.

Not only has this portion of the research served to build a case for the usefulness of the IRC framework as both a design and research aide for notification systems, it has revealed several promising directions for future work. While many directions were discussed in context throughout this chapter, three additional directions are discussed in the next chapter.

Chapter 8

INTEGRATIVE CASES, CONCLUSIONS, AND FUTURE WORK

The research described in this dissertation responds to the need within the human-computer interaction field to address ubiquitous and multitasking systems more scientifically, investigating the usefulness of a new research framework for a particular class of systems. *Notification systems* are interfaces typically used in a divided-attention, multitasking situation, attempting to deliver current, valued information through a variety of platforms and modes in an efficient and effective manner. This work has presented a unifying framework, the *IRC framework*, for understanding, classifying, analyzing, developing, evaluating, and discussing notification systems—focused on promoting scientific growth and knowledge reuse in research and design efforts.

Much of our research effort reported so far has developed the conceptual aspects of the IRC framework (Chapter 4) and tools to support it (Chapters 5 and 6). This work was a necessary precursor to studying the application of the framework to design practice and design research. In Chapter 7, we examined how the IRC framework (and its tools) should be integrated with a claims library to support design knowledge reuse. Testing indicated that the IRC framework should not merely be used as an indexing mechanism to access reusable design components; instead, we saw that automated support for design reuse should be situated within a design process. To this end, we developed a prototype of an integrated design environment, LINK-UP, to facilitate exploration and refinement of IRC metrics throughout the design process. As discussed in the previous chapter, preliminary testing of the LINK-UP processes was encouraging.

However, we have not yet presented an argument that the IRC framework, embodied within the LINK-UP system, is a valid design process enhancement. In this final chapter, we present three integrative case studies as a first step in this argument (Section 8.1). Then, we recap the conclusions and contributions made throughout our work (Section 8.2). Finally, we discuss directions for future work (Section 8.3) that should be taken to strengthen the argument we start here and broaden beneficial use of the IRC framework and LINK-UP.

8.1 Integrative Case Studies

In this section, we look at the results of actual design efforts in which the IRC framework and reusable design knowledge were used. The argument we would like to make is twofold: 1) a more useful and usable system results from the use of the IRC framework during a notification interface's design process, and 2) reusable knowledge related to successful notification system design is easily accumulated and leveraged with the use of LINK-UP. Certainly, this is not an easily supportable argument, since actual design efforts usually cannot be controlled for scientific observation and "success" is often a matter of opinion. Therefore, at this stage we look to collect anecdotal evidence in the form of case material relating to our intended outcomes.

In Section 5.2, we examined several real design efforts, those of both professionals and novices, to determine how often designers of notification systems reason about artifacts in terms of user interruption, reaction, and comprehension (IRC). We observed that professional designers expressed IRC-related design rationale more often than novices. Since our research products (i.e., the IRC framework, related tools, and LINK-UP) are designed to help novices consider notification system critical parameters, the case material in this section also involves novice design efforts.

We summarize three case studies of actual notification systems design projects, all of which include some component of design artifact reuse supported by a prototype notification claims library and IRC tools. Each case depicts real events and outcomes, as experienced by other designers within our research lab. Case observations were made about once per week on average, including discussion with designers, review of materials and testing results, and reflection on final reports. Throughout a narrative of each design process, we discuss implications for activity development within the claims library. Case observations exposed breakdowns in proposed activities, suggested alternate implementations, and validated specific parts of our intended research outcomes. Most of the material presented in this section was published in the 2005 *Proceedings of the Hawaii International Conference on System Sciences* [36].

Key focus points. In assembling these case studies, our purpose is to collect evidence indicating the IRC framework and LINK-UP are beneficial to a notification system design process and transfer of basic research. Specifically, we focus on building the case that the new system extends an novice designer's capability to put into practice many of the foundational HCI concepts reviewed in Chapter 2. This notion is expressed as our *key focus points*, practices grounded in HCI literature that we strive to transfer into a successful design process. We contend that these points will be identifiable if our design approach is valid:

1. **Systematic, tool-supported questioning of Norman's conceptual models—** As a specific "designer motivation" to reuse, LINK-UP helps designers establish a clear and distinct design model.

2. **Iterative critical parameter analysis to focus on psychological effects—** An analysis process that allow designers to *narrow focus of concern* to identify components of interest.

Developers need to be able to move from broad to detailed considerations of their requirements.

3. **Design knowledge abstraction with Sutcliffe's generic tasks and domain-specific critical parameters**—Designers need to be able to *index components in abstract terms*, allowing application to a broader range of requirements. By stripping the context from the initial presentation of content, the general underlying solution can be recognized by future developers.

4. **Inferential, fine-grained search indices through design model and design activity context** —To allow developers to relate to generic abstractions, use of design knowledge repository structures must *provide sufficient context*. Through use of generic building blocks (object models and task structures) designers must be able to judge details of the original context, helping them match it to their own. Though seemingly contradictory to the previous point, context, while not initially as important as other factors, in fact is important at progressive levels of component evaluation.

5. **Reusable knowledge that is archived as a by-product of design activity**—Repository solutions must also *account for the many barriers to knowledge management*, such as the overhead in preparing components to be reusable and the fast pace of technology change.

8.1.1 Case Study 1: Designing an IVIS input method

The first case describes a development process for an input method to an in-vehicle information system (for additional details see [63]). The intent of this design was to provide a competing technique to input methods like voice recognition or gesture systems to be used in the attentionally demanding driving situation. Although product development was discontinued after initial user testing, we view this as a successful design case because access to reusable knowledge helped expose design flaws before the development effort became costly.

Phase 0: Basic research. In his work, Chuck the researcher was eager to establish tradeoffs inherent in secondary task input methods, particularly those that could be used in vehicles for digital music selection. After several months of hardware and software development, the researcher performed a lab-based user test result comparison with a touch-screen interface, a voice recognition mockup, and a graffiti-like input method. The graffiti-like input method was shown to promote enhanced awareness required for complex menu traversal, thus decreasing distraction from the driving task. While the graffiti-like method required fairly high learn time, as an input method for digital music selection that would be generally marketed to a young demographic eager to adopt new technology, this was not determined to be a serious issue.

These psychological effects and the design decisions that caused them were recorded as claims within a prototype version of our claims library. Each claim was characterized by the research with an IRC rating describing the usability test outcomes, as well as the generic task(s) it supported (as defined in [118]).

Phase 1: Designing Thumb-type. Several months later, a new design initiative is begun by a novice designer (Chris):

> *Having heard that a colleague successfully developed a graffiti-like input method for in-vehicle digital music selection, Chris was eager to adapt the input method to allow selection of characters for a vehicle navigation system. As he began reasoning about specific requirements for the adapted input method, the designer turned to the claims library to consider the intended psychological effects and specific implementation settings, as well as to obtain benchmark usability criteria for his own user testing. The claims library contained claims about several different input methods, including the ones his colleague designed and compared his design against, as well as several others that were part of the collection.*

So far, the designer's activities with the design knowledge repository leverage basic design-science benefits present through a repository infrastructure.

> *Reflecting on the IRC ratings for each of these claims, he saw that no input method was able to achieve high reaction without also affecting either moderately high interruption or comprehension. In turn, the designer realized that the vehicle navigation system would be marketed to a broader demographic that would include users less willing to learn any complicated skill, suggesting that the graffiti-like input method would be inappropriate (it would cause unacceptably high interruption or reaction). The designer recognized an opportunity for developing an input method that could be used almost automatically, requiring very little attention or working memory access.*

At this point in the case, the designer has recognized an opportunity for innovation, rather than adaptation, encouragement that our claims library may be a step in the right direction toward supporting diverse design approaches. The designer's realization was spurred by consideration of the three-dimensional design space suggested by IRC factors. He was able to identify that the required IRC factors (forming the design model) were different those targeted or achieved by existing methods. Although this realization could have emerged from user surveys, marketing analysis, or even user testing, these types of processes would have been more costly than the analysis performed by the designer—validating key focus point #2.

> *Seeking inspiration for a new design to support automaticity (indicated by low interruption and high reaction goals) in the generic task of "selection," the designer continued to browse the claims library. Instead of just looking for knowledge related to design of input methods, this time he focused on abstract search indices that indicated the generic design task and desired IRC values. He conveyed confidence in this search strategy.*

Use of the abstract search indices provided some hope that key focus point #3 was being adequately supported. However, we note here that this designer was very experienced with Sutcliffe's generic task ontology.

> *Although several claims were returned by this search, the designer was disappointed that more claims were not available and that claim quality and evidence was inconsistent. Some claims summarized observations made in usability studies or presented in published papers, while the source of other claims was unclear.*

Here, we start to see some tensions resulting from barriers to knowledge management (key focus point #5). In the version of the claims library that this designer was using, the process of adding claims to the system was an action designers performed (or, more often, did not perform) during the documentation phase of their project. Although administrators sporadically enforced claim quality, no mechanisms existed to encourage quality or submission.

> *However, one of the claims returned sparked an idea for the designer. The claim made reference to providing selection capability "at the fingertips of a user." When the designer extended the analogy to an in-vehicle navigation system, he thought of steering-wheel mounted controls that would allow a user to type alpha-numeric characters with two 8-directional pads manipulable with a thumb (thus, Thumb-Type). Character mapping decisions were made to promote ease of learning through the most intuitive orientation-it was assumed that users would be able to select characters as easily when they were driving as they could when they were focusing on the selection task. The designer developed a simple prototype in a few days, adequate for conducting testing of users driving in a simulator.*
>
> *Based on the IRC benchmarks obtained with the original graffiti-like input method for the music selection task, user testing of Thumb-Type was discouraging. Actual user performance data indicated this particular implementation was not close enough to the design goals (supporting low interruption, high reaction, and low comprehension) to suggest continued development might lead to a valuable innovation. However, results and design rationale were archived as a point of comparison for alternate implementations, and the idea was brought to closure in a few weeks, rather than months.*

Case analysis. Unfortunately, the design process outcome in this case did not lead to a successful design product, but in a design-science context, it was a valuable process. A relatively inexperienced designer recognized an opportunity for innovation and the initial design was ruled out expediently. Furthermore, the knowledge resulting from the initial design effort (otherwise unworthy of publication due to its failure), once archived in the claims library, would prevent another designer from making the same incorrect design choices and assumptions. Most importantly, this case exposed strengths and weaknesses of the claims library for supporting the activities noted in a few of the key focus points. The next case study looks at a design facilitated by an improved version of the design knowledge repository.

8.1.2 Case Study 2: Designing a news notification system

Phase 0: Basic research. Prior to starting an interface design for a news-related notification system, we gathered results from lab-based testing of notification design elements, such as the results presented in Chapter 3. Results were expressed as claims that highlighted the psychological consequences introduced by a specific notification element in general terms of interruption, reaction, and comprehension effects. The general effects were also approximated in with IRC notation (three numeric values ranging from 0 to 1). This set of claims included the claims found at the end of Chapter 3, "*Quantitative comparison with position encoding*" and "*Quantitative comparison with color encoding.*"

Phase 1: Seven competing designs are compared. This case begins with a design challenge: competing groups were tasked to design and implement a news notification system.

> *As preliminary work for an interface development bid, seven design teams (which consisted of four or five members, including industrial systems engineers, programmers, and HCI specialists) developed rapid prototypes of a notification system that could deliver news-related information to desktop computer users. The interface was envisioned to be part of a subscription service for premium news feeds-a persistent desktop interface client would ensure that the subscriber (user) stayed aware of late-breaking information that was essential to him, and could readily access full versions of the news content. Another essential design requirement was ensuring that the system would not be annoying to a user during short or long-term use (through any unwanted distraction or interruption), since that could impact satisfaction with their subscription.*

Note that these user goals relate to desired (rough) levels of interruption, reaction, and comprehension, or IRC.

> *The seven development teams each pursued separate design proposals. They gathered detailed requirements through interview and focus group sessions with potential users, reflected on psychological tradeoffs for various design options, and developed limited-functionality prototypes (all seven designs displayed content from the same static source for a two-hour period of time—we were interested in comparing the effectiveness of the display techniques). A single testing team obtained performance metrics and subjective feedback for each prototype after it was used for several minutes by users engaged in other work tasks. Each prototype was also analyzed by three experts to determine its effectiveness at supporting the user goals. Strengths and weakness of each prototype, as related to specific dominant design features, were identified, recorded as claims, and placed in the claims library. Since multiple designs showed strong development potential, the decision was made to have a new developer design another option, attempting to reuse several of the strongest features from the initial prototypes (specific features to reuse were not specified).*

Phase 2: Iterative design of NewsBar Notification. As our research team was exploring how to integrate project archival services with design knowledge repository access points, we asked the new developer (Alan, the continued subject of Case 2) to perform specific activities that would allow reflection on LINK-UP modules before they were fully implemented. These activities were broadly specified as:

- Narrow development objectives to a specific design model (expressed in terms of critical parameters),
- Collect reusable design ideas and tradeoffs to support the design model,
- Build a prototype interface,
- Conduct testing to determine user's model critical parameter ratings.

These steps summarize the activities supported by LINK-UP.

First, the developer (an undergraduate programmer) proceeded through a few requirements analysis steps to narrow specification of the problem and facilitate its translation to abstract terms. After drafting scenarios to describe anticipated user interaction, the developer identified the generic tasks that users would perform and sequenced them as a hierarchical task analysis (HTA) within Norman's six stages of action (i.e., perceive, interpret, etc.)[100]. Next, he used IRCspec, the tool developed to help designers obtain specific IRC ratings for their design model (see Chapter 5).

After using the IRCspec tool in the Requirements Analysis module of LINK-UP, designers are provided with a template that suggests how each critical parameter (i.e. interruption) could be expected to change as the user interaction transitions through the stages of action. These types of features (an extension of key focus point #2) were designed to address key focus point #4 and should be most useful to users that have difficulty with abstracting their requirements or finding generic design knowledge that might be applicable to their needs.

Once this process was complete, the developer was pleased that he was thinking about the design in very thorough and high-level terms. He admitted that going through the process made him abandon an idea that he initially had when first hearing about the design assignment (recognizing that it would not be suitable). At this point, he had a framework of generic tasks and IRC ratings to guide his search for suitable claims. He was able to identify many claims that influenced his design, to include several from the initial prototypes, a few others that had come from both analogous and very different systems, as well as the "Quantitative comparison with color encoding" from our basic research.

This satisfactorily demonstrated how expanded design tools and services helped further key focus point #1 through clearer establishment of the design model.

> *After the developer prototyped a new notification system design (NewsBar), an expert evaluator predicted how well the system would support user goals by estimating a user's model IRC rating, (using an assessment technique for the IRC critical parameters described in Chapter 6). Comparing his design model with the user's model, the developer was able to instantly recognize that his design would not produce a high enough level of user reaction and trace the expected problem to specific claims and design features (which were included to produce appropriate reaction). He made several changes to both the design and his design model, noting that the combination of two claims, one pertaining to ticker rate and another related to relative size, could not be combined as easily as he first thought.*
>
> *Through further effort (the entire effort took less than a week of full-time work), the developer was able to produce a design that experts determined would meet the user goals. While inspired by elements of other systems, it bore little resemblance to any in the end.*

Here, we see validation for our key focus point #1 strategy, as well as some ideas for resolving knowledge management tensions (key focus point #5).

Case analysis. Reflecting on this case, the designer was able to craft a reliable notification system that would fulfill key user goals by synthesizing the experience of many other designers who had addressed similar problems. We expect that, through continued design work and design knowledge collection in a domain such as input methods for in-vehicle systems (the focus of Case 1), a designer would be able to follow the process illustrated in Case 2 for any domain to produce sound interface designs.

8.1.3 Case Study 3: Designing a notification system for ticket prices

While there is certainly some value in enabling a design process that reliably produces adequate design products (fulfilling basic requirements for usability and utility), we aspire to a loftier goal—toward supporting exceptional interface design. Case 3 demonstrates such a design product and reviews the system support that played an integral part in its development.

Phase 1: Spring Seminar individual projects. A very successful design project emerged from a 15-week undergraduate seminar activity in which seven computer science students interested in HCI were challenged to develop notification systems. Although the system was to deliver users information about airline flight prices that were available online, the instructors did not specify user goals that the system should support, but instead encouraged students to identify the critical parameter levels they thought were important, and to develop their systems accordingly. Students proceeded through the same requirements analysis steps identified in Case 2: developing problem scenarios and generic hierarchical task analysis, as well as using the IRC design model rating tool.

As the students embarked on their design work, group discussions showed that students had a variety of different conceptions about the important goals. It was somewhat disappointing that they tended to cover all of the bases with their designs rather than identifying tradeoffs among the three critical parameters. Although the individual designs that the students developed did not gravitate to any distinct system class within the notification systems design space, promising features began to emerge in each. Some of these features were identified through reusable claims in the claims library, while others were artifacts of the students' own invention. However, the most promising aspect of these features was that they embodied a strong tradeoff between the three parameters, even if the tradeoff was not supported by the system as a whole.

At this point in the case, the students were struggling to define strong design models (key focus point #1), but the reasoning process based on critical parameters was starting to provide the necessary focus for design improvement (key focus point #2). Recognizing this state, the instructor drew out discussion to highlight desired differences between the critical parameters students were targeting, often having students compare intentions. While some students were able to identify and reuse design artifacts successfully, they generally had more difficulty with the generic indices (key focus point #3) than designers in Case 1 and 2, but the recommendation features supporting key focus point #4 (described in Case 2 commentary) were not yet implemented within the IDE.

To help the students further realize some of the limitations in their designs, the instructor organized two usability evaluation processes, both using the analytic evaluation module in LINK-UP. First, students transferred design rationale and screenshot depictions into the claims library through the IDE. Then, other students acted as anonymous expert reviewers and, with the help of a user's model IRC assessment tool (IRCresults), they provided each other with user's model ratings. Next, the designers each used a tool within the system to prepare and conduct an empirical test with a few participants, obtaining performance metrics in a dual task situation that allowed calculation of an actual user's model IRC. As expected, none of the designers were not particularly pleased with their results.

By conducting these evaluations through LINK-UP, not only were the designers facilitated in receiving feedback, but the design expectations and actual results became a permanent part of the claims library—a natural knowledge byproduct of the design process. This strategy effectively addressed key focus point #5, especially through the capture of poor design decisions (still an extremely valuable source of knowledge) that would not have normally been archived.

While the students reflected upon their individual design intentions and products, they were given the freedom to continue their development efforts as they saw fit—they could revise individual design models and interfaces or work in teams to improve a more promising prototype.

Phase 2: Notifly. An upcoming undergraduate research symposium provided motivation to produce the highest quality system possible, since the top submissions would receive large cash prizes.

> *It was somewhat surprising that, with only a month left before the symposium, all seven students decided to work together on a completely new system (although still a flight price notification system). They reasoned that they had the best chance of creating a high-quality system by reusing features from several of the prototypes to support four distinct design models that would correspond to user customization options.*

Certainly, creating a new system from scratch was not the path of least resistance (especially working as a large team), nor a strategy that would position each individual with the chance for the largest cash award (any prize would have to be split in seven parts). However, the decision reflects the confidence students gained in the approach of selecting a distinct design model in terms of critical parameters and deliberately developing and reusing appropriate interface design components to match (key focus points #1 and 2).

> *As the team of students developed their new design, they drew from each other's claims, which were accessible through the claims library and supported with evaluation results. They made rapid progress and generated a new prototype, which they validated with user testing just in time for the symposium.*

We attribute much of the students' ability to rapidly organize their design goals and achieve consensus about interface decisions to the structure imposed by the design process. Continually questioning the design model IRCs when there was doubt about implementation options and referring to results obtained in earlier testing (apparent through the claims library) helped the students judge their own progress and readiness to move to new issues.

> *At the symposium, the resulting system, Notifly, was selected overwhelmingly by the approximately one hundred symposium attendees for the "People's Choice" award. In addition, the four groups of industry judges representing corporate program sponsors chose Notifly for the 2nd Place "Industry Choice" award.*

Case analysis. Few at the symposium would have guessed that the award-winning Notify had been developed in such a short iteration cycle, or had been based on such different initial prototypes. Although there were many factors that may have influenced this outcome, we believe that the most dominant factor was the use of the LINK-UP system. With this tool, the team was able to take advantage of reusable design knowledge and apply it in an innovative way that resonated with real people.

8.2 General Conclusions and Contributions

Throughout the chapters of this dissertation, we have argued that a new area of interface research can benefit from a more structured approach to design. We spoke of notification systems, but the particular system class may be thought of as an analogy to any new class of human-computer interfaces. Our argument began with a survey of the research activity within the notification systems field:

- Through literature review (Chapter 2), we recognized disjoint research and development efforts within the notification systems community.

By using extensible experimental platforms, we sought to offer a more comprehensive approach to building knowledge about notification information design. Our gedanken first approach, detailed in Chapter 3, produced *some* new design knowledge, but is more interesting as a testimony of the frustration in producing basic research and attempting to position the results for designers. Eager to discover an approach that would accelerate the transfer of basic research results into theory-based design practice,

- We articulated the need for a notification systems knowledge-organization taxonomy, capable of describing both user goals and design effects, while expressing the tension between attention and utility.

Our basic concept for this taxonomy, developed in Chapter 4 as the IRC framework, blends many new and old ideas within HCI literature. We hope to advance the already strong theme of redesigning to mitigate undesirable psychological effects, put forth by Carroll and others [28]. Borrowing Newman's notion of "meaningful and measurable" critical parameters [97], we gain a sharpened focus to describe the attention–utility tradeoff in generic, referential terms. Aspiring to create reusable claims, as Sutcliffe and Carroll demostrate [116], we first envisioned the taxonomy to be a faceted knowledge search mechanism—a more generic, numerical characterization of the critical tradeoffs expressed by the claim. However, themes from Norman's work [100] strongly influence us as well, suggesting an enlargement in the role of the IRC framework within a design process.

- We demonstrated use of the IRC framework for facilitating stage of action analysis (Section 4.2.3), and as a usability evaluation tool that helped explain intuition-based redesign decisions of expert notification designers (Section 4.3.2).

As we began to speculate on other design cycle efficiencies that the IRC framework might create, we carefully considered how the use of the framework could be fundamentally improved. Recognizing that an assessment of intended or actual values of the critical parameters is a highly complex and subjective process, we sought to develop supporting software tools. First, we reflected on our target users:

- Through an analysis of design rationale, we noted that a variety of expert notification system designers usually reasoned about IRC-related tradeoffs, while novice designers tended not to consider interactions among these psychological effects (see Section 5.2).

Designing our tools for the novice designer, we developed two software tools:

- The IRCspec tool (Section 5.3), a questionnaire-based dialog for specifying a design model IRC rating. We tested IRCspec to ensure that users obtain reasonably consistent IRC ratings for the same design problem context (Section 5.4).

- The IRCresults assessment techniques (demonstrated as both an analytic and empirical evaluation tool in Sections 6.4 and 6.5), which take as inputs the concrete, measurable effects of each critical parameter and outputs an abstract value, comparable with the abstract design model. The analytic application of IRCresults was shown to demonstrate sufficient interrater reliability and express meaningful usability characterizations.

Having these two tools available widened the possibilities for integrating the IRC framework with an iterative interface design process. Now practical were design reasoning activities that depend on consistent and accurate system description, interface evaluation, and design knowledge access and classification creation. Thus, our focus shifted to an exploration of implementation possibilities that would support such activities.

- We developed a prototype of a claims library, tailored the search indices for the notification design domain, and contributed initial design knowledge content (Section 7.1). We conducted formative testing on the claims library and observed several limitations with this initial approach.

- To merge critical parameter questioning and claims reuse activities with a typical design process, we developed the vision for LINK-UP (Section 7.2), a computer-aided design support system.

As an integrated design environment, the modules of LINK-UP that we specified will help a designer conduct activities like requirements analysis and analytic evaluation. Since LINK-UP modules include the IRC tools and access an underlying claims library, we have operationalized the enlarged IRC framework role first suggested in Section 4.2.3. While this portion of the research is still in formative stages, we have begun collecting evidence in a limited validation of LINK-UP's benefits.

- Initial testing of the LINK-UP modules showed the system to be comprehensible to designer-users, and our documentation is sufficient for continued implementation.

- Case studies of novice designers (see Section 8.1) who used the IRC and/or LINK-UP in their design process experienced successful design results. Their design processes exhibited desirable characteristics, consistent with design theory.

The capabilities introduced by LINK-UP suggest many new directions for notification system development support tools. This encourages the thought that when valuable design tools embody critical parameters and are coupled with readily accessible reusable design knowledge, interface development will improve as a scientific endeavor.

Resources for future work offered by this research include a valuable and living implementation of a notification systems component claims library, a tool-supported design assessment system for notification system conceptual models, and development of a research agenda toward a science of design centered on design rationale examination. The groundwork established here creates many new research opportunities for other students and researchers within the HCI community.

8.3 Directions for Future Work

An essential goal of this research program is to provide a foundation that will allow many valuable directions for future work. Our research was intentionally designed to contribute a vital portion of the groundwork necessary for integrating the IRC framework within an integrated design process—the more interesting future work involves the application of our ideas. Many directions for future work have been discussed in context within this dissertation, for example, Section 7.1.5 outlines six projects that could enhance the core claims library within the LINK-UP system. In this final chapter, we highlight the directions that should be taken next to broaden the impact of the IRC framework and LINK-UP, new questions and opportunities that can guide several years of continued research.

8.3.1 Educating novice designers with LINK-UP

Arguments from a formative workshop to define a new software-systems initiative within the National Science Foundation, *Science of Design*, help frame the needs of tomorrow's HCI students and educators. Many authors note the changing quality of design work in general-transition from a largely individual or small group activity to a distributed, interdisciplinary team effort often involving re-development of an existing system. Nielsen predicts that the future trend of software development will involve offshore team implementation efforts guided by domestic user research and design work. Based on this prediction, distance between design and development functions will increase further, posing new challenges to designing usable and useful systems [99].

If we are serious about advancing HCI as a design-science, we must start integrating these considerations into our HCI and software development educational programs. Educators preparing students for software design professions must be poised to meet this challenge, teaching new approaches that help students practice distributed design efforts. It is our thought that LINK-UP can facilitate these challenges. We have already observed that structuring HCI education around the IRC framework can have many distinct benefits, as observed in [35]. As we work to integrate LINK-UP into HCI education, other arguments center on three important themes that will guide our continued development.

- The first theme is an argument for *design methods that support an improved understanding of problem spaces*. As noted by Brooks, "often the hardest part of design is deciding what to design," since designers often lack a precise description of the problem to be solved [17].
- Brooks also introduces a second theme for software development processes that embodies a design-science vision—*tools must be available to present detailed option sets for design choices, ideally that assist in co-evolution of the problem as well as the design solution* [17]. Likewise, Shaw argues for systematic guidance of design decisions, specifically those that express costs and benefits of software design and help designers consider user preferences [110]. Potts argues that knowledge should be accumulated by recording the "science of the designed" through artifact-as-phenomena investigations, modeled as pattern abstraction [102].
- The third theme elaborates on Carroll's argument that a commitment to universal usability and responsible selection of social impacts requires *proactive and formative engagement of designers* as an integral part of design conception and development" [25]. Fischer identifies domain-oriented design environments as an essential facet for supporting "reflection in action," asserting that these environments should integrate specification components, critiquing components, case-based libraries, and evolving artifacts [50].

We must consider how to weave these themes into educational materials for HCI instruction. While LINK-UP can provide an environment for theory-based development work, new educational activities must be designed to address the ever-changing needs of the software development profession.

8.3.2 Synthesizing multidisciplinary research results

A second direction for future work with LINK-UP has also emerged from the Science of Design workshop. In her position statement, John includes many other considerations related to research in design teamwork, to include process and tool support for interdisciplinary team members, helping to transfer knowledge from the behavioral sciences for prediction of design idea feasibility prior to extensive building or prototyping efforts [70]. She argues that, in order to enable multidisciplinary design work, the interested party (in our case, the notification system designer) must do most (95%) of the work by building the processes and infrastructure that allow the transfer of knowledge.

To this end, we have begun to think about how LINK-UP can be used by other local researchers from other disciplines (e.g., from departments like Industrial and Systems Engineering, Psychology, Sociology, Education, and the School of the Arts). That is, once the tool is in place and initial design efforts are available for demonstration, we will organize events that inform potential research partners about the opportunities available for collaboration in interface design research, made possible by LINK-UP. Some of these opportunities will include use of the LINK-UP system and design knowledge content in other classes, perhaps to support a special interdisciplinary design/testing activity or research assignment. Other opportunities might involve deeper investigation of specific design approaches (i.e., use of interactive public displays or haptic interfaces) or application areas (i.e., community computing, interfaces for disabled users, or groupware), facilitated by web-based access to a repository of claims and testing results. We also expect researchers from other disciplines will have standard design and testing methods that should be integrated into LINK-UP's tool suite, another direction for follow-on efforts.

8.3.3 Advancing notification systems design with a reference task agenda

With the capability to accurately and consistently compare design and user's models, as well as abstract user tasks into generic and generalized tasks, we have established a research framework that is ready to gauge design progress toward reference tasks.

Whittaker, Terveen, and Nardi present an excellent argument for a reference task research agenda within HCI [133]. They propose an agenda that provides a structured approach to identifying design requirements and sharing research findings through empirical analysis of essential user tasks, systematic analysis of user interaction, and comparison of measured task performance. *References tasks* are common user tasks within a research field that can be described by general problem definitions and user requirements, measured by standard experimental tasks, datasets, contextual information, and metrics, but instantiated by an unlimited number of design variations.

Within our research, we have focused on the utility that comes from comparing a single system's design model to the user's model, especially as part of an iterative design process. However, we can also speculate that comparing multiple systems to a single design model with the IRC framework would be useful. This will allow designers to identify which design provides best support for a given task and design model. Over time, benchmarks described by IRC values and usability metrics can develop for various notification tasks. This can help researchers distinguish difficult design problems that are in need of concerted research efforts (which would be indicated by design models that have few close user's model artifacts). Similarly, if a designer was able to compare his design to an existing design that was closer to the same design model, associated design history information (i.e. the iterative changes that brought the design to that point) could be useful for planning a cost-benefit analysis of design reengineering. All of these potential sources of utility can result from established reference tasks.

8.3.4 Narrowing the HCI–Software Engineering gap

Finally, as we lay the groundwork to further establish HCI as design-science, we must also strengthen the connections to software development practice. Although our strain of research has avidly promoted the merits of design reuse in the form of claims, developers will also desire more "tangible" reusable components. Connecting notification design to notification implementation activities within LINK-UP is a critical step. Claims can be associated with their corresponding software classes, and task models can transition to interactor diagrams and class hierarchies. Ideally, developers should be able to swap out portions of code within a project's development environment and instantly understand the anticipated differences in terms of critical parameters/psychological effects.

As the distinctions and complementary roles of patterns and claims are explored, the knowledge stored within LINK-UP may be able to be augmented with existing, external knowledge repositories. Perhaps, this is one of the most important projects to make progress on, if student programmers and software developers are expected to appreciate the notification systems development processes we introduce.

8.4 Closing

In summary, an overarching argument put forth by this dissertation is that interface design improvement will result from a more systematic, deductive approach to advancing the HCI body of knowledge. First, we must facilitate the transfer of knowledge from basic researchers (i.e., psychologists, sociologists, industrial engineers) to software developers, who combine the guidelines and principles developed through basic research into a working software system. In turn, the use of systems provide opportunities for reflection on the basic theories, which should transfer back to appropriate research activities.

To leverage an improved transfer of basic research, we need a more thorough understanding of the design spaces we hope to advance. This understanding must convey to a community of researchers the common problems where solutions exist, as well as those that still require innovative design initiatives–thus, allowing comparison of systems, suggestions for reuse, and recognition of progress.

With computer-aided design support systems that promote these goals, we can realize the benefits that emerge from all well-established science disciplines: a practical, value-adding engineering process. That is, through the structures on which we base our science, we should also be able to develop new interface products reliably, cost-efficiently, and with higher quality.

References

[1] Gregory D. Abowd and Elizabeth D. Mynatt. Charting past, present, and future research in ubiquitous computing. *Transactions on Human-Computer Interaction*, 7(1):29–58, 2000.

[2] M. J Adams, Y. K. Tenney, and R. W. Pew. Situation awareness and the cognitive management of complex systems. *Human Factors*, 37:87–104, 1995.

[3] C. Alexander, S. Ishikawa, and M. Silverstein. *A pattern language*. Oxford: Oxford University Press, 1977.

[4] J. R. Anderson and C. Liebiere. *The Atomic Components of Thought*. Lawrence Erlbaum Associates, Inc., Mahwah, New Jersey, 1998.

[5] Arthur Aron and Elaine Aron. *Statistics for Psychology*. Prentice Hall, Upper Saddle River, New Jersey, third edition, 2003.

[6] Allan Baddelly. Exploring the central executive. *The Quarterly Journal of Experimental Psychological*, 49A:5–28, 1996.

[7] Brian P. Bailey, Joseph A. Konstan, and John V. Carlis. The effects of interruptions on task performance, annoyance, and anxiety in the user interface. In *Proceedings of the IFIP TC.13 International Conference on Human-Computer Interaction (INTERACT 2001)*, pages 593–601, Tokyo, Japan, July 2001.

[8] J. A. Ballas, C. L. Heitmeyer, and M. A. Perez. Evaluating two aspects of direct manipulation in advanced cockpits. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '92)*, pages 127–134, Monterey, CA, 1992.

[9] Philip Barnard and Jon May. Representing cognitive activity in complex tasks. *Human-Computer Interaction*, 14:93–158, 1999.

[10] Philip Barnard and Jon May. Systems, interactions, and macrotheory. *ACM Transactions on Computer-Human Interaction*, 7(2):222–262, June 2000.

[11] Linda Ruth Bartram. *Enhancing Information Visualization with Motion*. PhD thesis, Simon Fraser University, Canada, 2001.

[12] Lyn Bartram. Enhancing visualizations with motion. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '98)*, pages 13–16, Raleigh, NC, 1998.

[13] Lyn Bartram, Colin Ware, and Tom Calvert. Moving icons: Detection and distraction. In *Proceedings of the IFIP TC.13 International Conference on Human-Computer Interaction (INTERACT 2001)*, Tokyo, Japan, July 2001.

[14] Michelle E. Bayles. Designing online banner advertisements: should we animate? In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '02)*, pages 363–366. ACM Press, 2002.

[15] Brandon Berry. Adapting heuristics for notification systems. In *Proceedings of 41st Annual ACM Southeast Conference*, pages 144–149, 2003.

[16] J. Bertin. *Semiology of Graphics*. University of Wisconsin Press, Madison, Wis, 1983.

[17] Fred Brooks. Is there a design of design?, November 2003. Position paper for the NSF Invitational Workshsop on Science of Design: Software-Intensive Systems, Available at: http://www.cs.virginia.edu/ sullivan/sdsis/.

[18] S. M. Brown, E. Santos, S. B. Banks, and M. E. Oxley. Using explicit requirements and metrics for interface agent user model correction. In *Proceedings of the Second International Conference on Autonomous Agents*, pages 1–7, Minneapolis, MN, May 1998.

[19] J. J. Cadiz, Gina Venolia, Gavin Jancke, and Anoop Gupta. Designing and deploying an information awareness interface. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 314–323. ACM Press, 2002.

[20] JJ Cadiz, Mary Czerwinski, D. Scott McCrickard, and John Stasko. Workshop: Providing elegant peripheral awareness. In *Conference Extended Abstracts on Human Factors in Computer Systems (CHI '03)*. ACM Press, 2003.

[21] Jonathan J. Cadiz, Susan R. Fussell, Robert E. Kraut, F. Javier Lerch, and William L. Scherlis. The awareness monitor: A coordination tool for asynchronous, distributed work teams. Available from www.cs.cmu.edu/ kraut, 2000.

[22] Stuart K. Card, Jock D. Mackinlay, and George G. Robertson. The design space of input devices. In *Conference Companion of the ACM Conference on Human Factors in Computing Systems (CHI '90)*, pages 117–124, 1990.

[23] John M. Carroll. Making use: a design representation. *Communications of the ACM*, 37(12):29–35, December 1992.

[24] John M. Carroll. *Making Use: Scenario-Based Design of Human-Computer Interactions*. The MIT Press, Cambridge, MA, 2000.

[25] John M. Carroll. Activity as the object of design, November 2003. Position paper for the NSF Invitational Workshsop on Science of Design: Software-Intensive Systems, Available at: http://www.cs.virginia.edu/ sullivan/sdsis/.

[26] John M. Carroll and Wendy A. Kellogg. Artifact as theory-nexus: hermeneutics meets theory-based design. In *Conference Companion of the ACM Conference on Human Factors in Computing Systems (CHI '89)*, pages 7–14, 1989.

[27] John M. Carroll, R. L. Mack, Scott Robertson, and Mary Beth Rosson. Binding objects to scenarios of use. *International Journal of Human-Computer Studies*, 41:243–276, 1994.

[28] John M. Carroll and Mary Beth Rosson. Getting around the task-artifact cycle: how to make claims and design by scenario. *ACM Transactions on Information Systems (TOIS)*, 10(2):181–212, April 1992.

[29] John M. Carroll, Mark K. Singley, and Mary Beth Rosson. Integrating theory development with design evaluation. *Behavior and Information Technology*, 11:247–255, 1992.

[30] Stephen M. Casner. Task-analytic approach to the automated design of graphic presentations. *ACM Transactions on Graphics*, 10(2):111–151, April 1991.

[31] J. D. Chase, Robert Schulman, H. Rex Hartson, and Deborah Hix. Development and evaluation of a taxonomical model of behavioral representation techniques. In *Conference Companion of the ACM Conference on Human Factors in Computing Systems (CHI '94)*, pages 159–165, 1994.

[32] C. M. Chewar, Edwin Bachetti, D. Scott McCrickard, and John E. Booker. Automating a design reuse facility with critical parameters: Lessons learned in developing the LINK-UP systems. In *Proceedings of the Conference on Computer-Aided Design of User Interfaces (CADUI)*, Funchal, Island of Madeira, Portugal, 2004.

[33] C. M. Chewar and D. Scott McCrickard. Adapting UEMs for notification systems. Workshop position paper, UbiComp 2002, Available at: http://csgrad.cs.vt.edu/ cchewar/paper7.pdf, 2002.

[34] C. M. Chewar and D. Scott McCrickard. Dynamic route descriptions: Tradeoffs by usage goals and user characteristics. In *Proceedings of the 2nd International Symposium on Smart Graphics*, Hawthorne, NY, 2002.

[35] C. M. Chewar and D. Scott McCrickard. Educating notice developers of notification systems: Targeting user-goals with a conceptual framework. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA 2003)*, Honolulu, Hawaii, 2003.

[36] C. M. Chewar and D. Scott McCrickard. Links for a human-centered science of design: Integrated design knowledge environments for a software development process. In *Proceedings of the Hawaii International Conference on System Sciences (HICSS 2005)*, Waikoula Village, Hawaii, 2005.

[37] C. M. Chewar, D. Scott McCrickard, Ali Ndiwalana, Chris North, Jon Pryor, and David Tessendorf. Secondary task display attributes: Optimizing visualizations for cognitive task suitability and interference avoidance. In *Proceedings of the Symposium on Data Visualization (VisSym '02)*, pages 165–171, Barcelona, Spain, 2002. Eurographics Association.

[38] C. M. Chewar, D. Scott McCrickard, and Alistair Sutcliffe. Unpacking critical parameters for interface design: Evaluating notification systems with the IRC framework. In *Proceedings of the ACM Conference on Designing Interactive Systems (DIS '04)*, Boaston, MA, 2004.

[39] William S. Cleveland. *The Elements of Graphing Data*. Hobart Press, Summit, New Jersey, 1994.

[40] William S. Cleveland and Robert McGill. Graphical perception: Theory, experimentation, and application to the development of graphical models. *Journal of the American Statistical Association*, 79(387), September 1984.

[41] Edward Cutrell, Mary Czerwinski, and Eric Horvitz. Notification, disruption, and memory: Effects of messaging interruptions on memory and performance. In *Proceedings of the IFIP TC.13 International Conference on Human-Computer Interaction (INTERACT 2001)*, pages 263–269, Tokyo, Japan, July 2001.

[42] Andrew Dahley, Craig Wisneski, and Hiroshi Ishii. Water lamp and pinwheels: Ambient projection of digital information into architectural space. In *Proceedings of the Conference on CHI 98 Summary : Human Factors in Computing Systems*, pages 269–270. ACM Press, 1998.

[43] Gert de Haan. ETAG-based design: User interface design as mental model specification. In P. Palanque and D. Benyon, editors, *Critical Issues in User Interface Systems Engineering*, pages 81–92. Springer Verlag, London, 1996.

[44] Peter Denning and Pamela Dargan. Action-centered design. In Terry Winograd, editor, *Bridging Design to Software*, pages 105–119. ACM Press, 1996.

[45] Robert L. Duchnicky and Paul A. Kolers. Readability of text scrolled on visual display terminals as a function of window size. *Human Factors*, 25(6):683–692, 1983.

[46] Mica R. Endsley, Betty Bolte, and Debra G. Jones. *Designing for Situation Awareness: An Approach to User-Centered Design*. Taylor & Francis, New York, 2003.

[47] James T. Enns and Ronald A. Rensink. Preattentive recovery of three-dimensional orientation from line drawings. *Psychological Review*, 98:335–351, 1991.

[48] K. Anders Ericsson and Walter Kintsch. Long term working memory. *Psychological Review*, 102(2):211–245, 1995.

[49] Alan Fabian, David Felton, Melissa Grant, Cyril Montabert, Kevin Pious, Nima Rashidi, Anderson Ray Tarpley III, Nicholas Taylor, C. M. Chewar, and D. Scott McCrickard. Designing the claims reuse library: Validating classification methods for notification systems. In *Proceeding of the 41st Annual ACM Southeast Conference (ACM-SE 2004)*, pages 357–362, April 2004.

[50] Gerhard Fischer. Coping with software intensive systems–design, design communities, meta-design and social creativity., November 2003. Position paper for the NSF Invitational Workshsop on Science of Design: Software-Intensive Systems, Available at: http://www.cs.virginia.edu/ sullivan/sdsis/.

[51] K. I. Foster. Visual perception of rapidly presented word sequences of varying complexity. *Perception and Psychophysics*, 8:215–221, 1970.

[52] Michael M. Granaas, Timothy D. McKay, R. Darrell Laham, Lance D. Hurt, and James F. Juola. Reading moving text on a CRT screen. *Human Factors*, 26(1):97–104, 1984.

[53] D. M. Green and J. A. Swets. *Signal Detection Theory and Psychophysics*. Wiley, New York, 1966.

[54] Paul Green. The 15-second rule for driver information systems. In *Proceedings of the ITS America Ninth Annual Meeting*, pages CD–ROM, Washington, DC, 1999.

[55] Saul Greenberg and Chester Fitchett. Phidgets: Easy development of physical interfaces through physical widgets. In *Proceedings of the ACM Conference on User Interface Software and Technology (UIST '01)*, Orlando, FL, November 2001.

[56] Saul Greenberg and Michael Rounding. The notification collage: Posting information to public and personal displays. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '01)*, pages 515–521, Seattle, WA, April 2001.

[57] Beverly L. Harrison, Hiroshi Ishii, Kim J. Vicente, and William A. S. Buxton. Transparent layered user interfaces: An evaluation of a display design to enhance focused and divided attention. In *Conference proceedings on Human factors in computing systems (CHI '95)*, pages 317–324. ACM Press/Addison-Wesley Publishing Co., 1995.

[58] Christopher G. Healey, Kellogg S. Booth, and James T. Enns. High-speed visual estimation using preattentive processing. *ACM Transactions on Human Computer Interaction*, 3(2):107–135, 1996.

[59] Christopher G. Healey and James T. Enns. Large datasets at a glance: Combining textures and colors in scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):145–167, 1999.

[60] Jeremy M. Heiner, Scott E. Hudson, and Kenichiro Tanaka. The information percolator: Ambient information display in a decorative object. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '99)*, pages 141–148, Asheville, NC, November 1999.

[61] Deborah Hix and H. Rex Hartson. *Developing User Interfaces: Ensuring Usability Through Product and Process*. John Wiley and sons, New York, NY, 1993.

[62] E. Hoadley. Investigating the effects of color. *Communications of the ACM*, 33(2):120–139, February 1990.

[63] Chuck Holbrook. *Input methods for Notification Systems: A Design Analysis Technique with a Focus on Input for Dual-Task Situations*. PhD thesis, Virginia Tech, Blacksburg, VA, 2003.

[64] Eric Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '99)*, pages 159–166, Pittsburgh, PA, May 1999.

[65] Eric Horvitz and J. Apicable. Learning and reasoning about interruption. In *Proceedings of the Fifth International Conference on Multimodal Interfaces*, Vancouver, Canada, November 2003.

[66] Eric Horvitz, Andy Jacobs, and David Hovel. Attention-sensitive alerting. In *Conference on Uncertainty and Artificial Intelligence (UAI '99)*, pages 305–313, Stockholm, Sweden, July 1999.

[67] Eric Horvitz, Carl Kadie, Tim Paek, and David Hovel. Models of attention in computing and communication: from principles to applications. *Communications of the ACM*, 46(3):52–59, 2003.

[68] Hiroshi Ishii and Brygg Ulmer. Tangible bits: Towards seamless interfaces between people, bits, and atoms. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '97)*, pages 234–241, Atlanta, GA, March 1997.

[69] Hiroshi Ishii, Craig Wisneski, Scott Brave, Andrew Dahley, Matt Gorbet, Brygg Ullmer, and Paul Yarin. ambientROOM: Integrating ambient media with architectural space. In *Proceedings of the Conference on CHI 98 Summary : Human Factors in Computing Systems (CHI '98)*, pages 173–174. ACM Press, 1998.

[70] Bonnie E. John. Position statement of bonnie e. john for science of design: Software-intensive systems, November 2003. Position paper for the NSF Invitational Workshsop on Science of Design: Software-Intensive Systems, Available at: http://www.cs.virginia.edu/ sullivan/sdsis/.

[71] James F. Juola, Nicklas J. Ward, and Timothy McNamara. Visual search and reading of rapid serial presentations of letter strings, words, and text. *Journal of Experimental Psychology: General*, 111(2):208–227, 1982.

[72] T. Jin Kang and Paul Muter. Reading dynamically displayed text. *Behaviour and Information Technology*, 8(1):33–42, 1989.

[73] Victor Kaptelinin, Bonnie A. Nardi, and Catriona Macaulay. The Activity Checklist: A tool for representing the space of context. *Journal of Experimental Psychology: General*, 111(2):208–227, 1982.

[74] D. E. Kieras and D. E. Meyer. An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction*, 12(4):391–438, 1997.

[75] Jason Chong Lee, Sirong Lin, C. M. Chewar, and D. Scott McCrickard. From chaos to cooperation: Teaching analytic evaluation with LINK-UP. In *Submitted to the Proceedings of the World Conference on E-learning in Corporate, Government, Healthcare, & Higher Education (E-learn '04)*, page 8 pgs, Washington, DC, November 2004.

[76] John D. Lee, Cher Carney, Steven M. Casey, and John L. Campbell. In-vehicle display icons and other information elements: Preliminary assessment of visual symbols. Technical Report FHWA-RD-99-196, United States Federal Highway Administration, December 1999.

[77] J. Lohse. A cognitive model for the perception and understanding of graphs. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '91)*, pages 137–144, 1991.

[78] Blair MacIntyre, Elizabeth D. Mynatt, Stephen Voida, Klaus M. Hansen, Joe Tullio, and Gregory M. Corso. Support for multitasking and background awareness using interactive peripheral displays. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology (UIST '01)*, pages 41–50. ACM Press, 2001.

[79] Jock D. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2):110–141, April 1986.

[80] Paul P. Maglio and Christopher S. Campbell. Tradeoffs in displaying peripheral information. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2000)*, The Hague, the Netherlands, April 2000.

[81] Lena Mamykina, Elizabeth Mynatt, and Michael A. Terry. Time aura: Interfaces for pacing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*, pages 144–151. ACM Press, 2001.

[82] Jennifer Mankoff, Anind Dey, Gary Hsieh, J. Kientz, Scott Lederer, and M. Ames. Heuristic evaluation of ambient displays. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI '03)*. ACM Press, 2003.

[83] D. Scott McCrickard. Maintaining information awareness with Irwin. In *Proceedings of the World Conference on Educational Multimedia/Hypermedia and Educational Telecommunications (ED-MEDIA '99)*, Seattle, WA, June 1999.

[84] D. Scott McCrickard, Richard Catrambone, C. M. Chewar, and John T. Stasko. Establishing tradeoffs that leverage attention for utility: Empirically evaluating information display in notification systems. *International Journal of Human-Computer Studies*, 8(5):547–582, May 2003.

[85] D. Scott McCrickard, Richard Catrambone, and John T. Stasko. Evaluating animation in the periphery as a mechanism for maintaining awareness. In *Proceedings of the IFIP TC.13 International Conference on Human-Computer Interaction (INTERACT 2001)*, pages 148–156, Tokyo, Japan, July 2001.

[86] D. Scott McCrickard and C. M. Chewar. Designing attention-centric notification systems: Five HCI challenges. In Chris Forsythe, Michael L. Bernard, and Timothy E. Goldsmith, editors, *Cognitive Systems: Human Cognitive Models in System Design*. in press.

[87] D. Scott McCrickard and C. M. Chewar. Attuning notification design to user goals and attention costs. *Communications of the ACM*, 46(3):67–72, March 2003.

[88] D. Scott McCrickard, C. M. Chewar, Jacob P. Somervell, and Ali Ndiwalana. A model for notification systems evaluation—Assessing user goals for multitasking activity. *Transactions on Computer-Human Interaction*, 10(4):312–338, December 2003.

[89] D. Scott McCrickard, C. M. Chewar, Jacob P. Somervell, and Ali Ndiwalana. A model for notification systems evaluation—Assessing user goals for multitasking activity. *Transactions on Computer-Human Interaction*, 10(4):312–338, December 2003.

[90] D. Scott McCrickard, Mary Czerwinski, and Lyn Bartram. Introduction: Design and evaluation of notification system interfaces. *International Journal of Human-Computer Studies*, 8(5):509–514, May 2003.

[91] Daniel C. McFarlane. *Interruption of People in Human-Computer Interaction*. PhD thesis, George Washington University, Washington DC, 1998.

[92] Daniel C. McFarlane. Comparision of four primary methods for coordinating the interruption of people in human-computer interaction. *Human Computer Interaction*, 17(3), 2002.

[93] H. Mori and Y. Hayashi. Visual interference with user's tasks on multi-window systems. In *Proceedings of the Fifth International Conference on Human-Computer Interaction*, volume 2, pages 80–85, 1993.

[94] Ali Ndiwalana, Nithiwat Kampanya, Ian McEwan, C. M. Chewar, D. Scott McCrickard, and Kevin Pious. A tool for participatory negotiation: LINKing-UP participatory design and design knowledge reuse. In *Participatory Design Conference (PDC 2004)*, page 4 pgs, July 2004.

[95] A. Newell. *Unified Theories of Cognition*. Harvard University Press, Cambridge, MA, 1990.

[96] William Newman. A preliminary analysis of the products of hci research, using pro forma abstracts. In *Conference Proceedings on Human Factors in Computing Systems : Celebrating Interdependence (CHI '94)*, pages 278–284. ACM Press, 1994.

[97] William M. Newman. Better or just different? On the benefits of designing interactive systems in terms of critical parameters. In *Proceedings of the Conference on Designing Interactive Systems : Processes, Practices, Methods, and Techniques (DIS '97)*, pages 239–245. ACM Press, 1997.

[98] William M. Newman, Alex S. Taylor, Christopher R. Dance, and Stuart A. Taylor. Performance targets, models and innovation in interactive systems design. In *Proceedings of the Conference on Designing Interactive Systems : Processes, Practices, Methods, and Techniques (DIS '00)*, pages 381–387. ACM Press, 2000.

[99] Jakob Nielsen. Getting the science of design used, November 2003. Position paper for the NSF Invitational Workshsop on Science of Design: Software-Intensive Systems, Available at: http://www.cs.virginia.edu/ sullivan/sdsis/.

[100] Donald A. Norman. Cognitive engineering. In Donald A. Norman and Stephen W. Draper, editors, *User Centered System Design: New Perspectives on Human Computer Interaction*, pages 31–62. Lawrence Erlbaum Associates, 1986.

[101] Catherine Payne, C. F. Allgood, C. M. Chewar, Chuck Hollbrook, and D. Scott McCrickard. Generalizing interface design knowledge: Lessons learned from developing a claims library. In *2003 IEEE International Conference on Information Reuse and Integration (IRI 03)*, pages 362–369, Las Vegas, NV, October 2003.

[102] Colin Potts. Policy modularity: Toward a science of socially-embedded system design, November 2003. Position paper for the NSF Invitational Workshsop on Science of Design: Software-Intensive Systems, Available at: http://www.cs.virginia.edu/ sullivan/sdsis/.

[103] Johan Redström, Tobias Skog, and Lars Hallnäs. Informative art: Using amplified artworks as information displays. In *Proceedings of DARE 2000 on Designing Augmented Reality Environments*, pages 103–114. ACM Press, 2000.

[104] Bradley Rhodes and Pattie Maes. Just-in-time information retrieval agents. *IBM Systems Journal*, 39(3–4):685–704, 2000.

[105] Kirsten Risden, Mary Czerwinski, Stephanie Worley, Lynda Hamiliton, Joe Kubiniec, Hunter Hoffman, Nancy Mickel, and Elizabeth Loftus. Interactive advertising: Patterns of use and effectiveness. In *Proceeding of the CHI 98 Conference on Human Factors in Computing Systems*, pages 219–224. ACM Press, 1998.

[106] I. Rock and A. Mack. Attention and perceptual organization. In S. Ballesteros, editor, *Cognitive Approaches to Human Perception*. Lawrence Erlbaum Associates, 1994.

[107] Robert D. Rogers and Stephen Monsell. Cost of a predictable switch between simple cognitive tasks. *Journal of Experimental Psychology: General*, 124(2):207–231, 1995.

[108] Mary Beth Rosson and John M. Carroll. *Usability Engineering: Scenario-Based Development of Human-Computer Interaction*. Morgan Kaufman, New York, NY, 2002.

[109] Judi E. See, Steven R. Howe, Joel S. Warm, and William Dember. Meta analysis of the sensitivity decrement in vigilance. *Psychological Bulletin*, 117(2):230–249, 1995.

[110] Mary Shaw. Research questions for foundations of complex software design, November 2003. Position paper for the NSF Invitational Workshsop on Science of Design: Software-Intensive Systems, Available at: http://www.cs.virginia.edu/ sullivan/sdsis/.

[111] T. Sheridan. Human factors of driver-vehicle interaction in the IVHS environment. Technical Report DOT HS 807 837, National Technical Information Service, Springfield MA, 1991.

[112] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Conference on Visual Languages*, pages 336–343, 1996.

[113] Jacob Somervell. Comparing heuristic evaluation methods for large screen information exhibits. Ph.D. Proposal, 2002.

[114] Jacob Somervell, C. M. Chewar, and D. Scott McCrickard. Graphical vs. textual secondary displays for information notification. In *Proceedings of the 40th Annual ACM Southeast Conference*, pages 153–160, April 2002.

[115] Jacob Somervell, C. M. Chewar, D. Scott McCrickard, and Ali Ndiwalana. Enlarging usability for ubiquitous displays. In *Proceeding of the 41st Annual ACM Southeast Conference (ACM-SE 2003)*, pages 24–29, March 2003.

[116] A. G. Sutcliffe and J. M. Carroll. Designing claims for reuse in interactive systems design. *International Journal of Human-Computer Studies*, 50:213–241, 1999.

[117] Alistair Sutcliffe. On the effective use and reuse of HCI knowledge. *ACM Transactions on Computer-Human Interaction*, 7(2):197–221, June 2000.

[118] Alistair Sutcliffe. *The Domain Theory: Patterns for Knowledge and Software Reuse*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, 2002.

[119] Alistair Sutcliffe and Peter Faraday. Designing presentation in multimedia interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '94)*, pages 92–98. ACM Press, 1994.

[120] David Tessendorf, C. M. Chewar, Ali Ndiwalana, Jon Pryor, D. Scott McCrickard, and Chris North. An ordering of secondary task display attributes. In *Conference Extended Abstracts on Human Factors in Computing Systems (CHI '02)*, pages 600–601. ACM Press, 2002.

[121] Peter Tolmie, James Pycock, Tim Diggins, Allan MacLean, and Alain Karsenty. Unremarkable computing. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '02)*, pages 399–406. ACM Press, 2002.

[122] D. Tufano, H. Knee, and P. Spelt. In-vehicle signing functions and systems concepts. In *Proceedings of the 29th International Symposium on Automotive Technology and Automation (ISATA) Dedicated Conference on Global Deployment of Advanced Transportation Telematics/ITS*, pages 97–104, Florence Italy, June 1996.

[123] Edward R. Tufte. *Envisioning Information*. Graphics Press, Cheshire, CT, 1990.

[124] Edward R. Tufte. *Visual Explanations*. Graphics Press, Cheshire, CT, 1997.

[125] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, 2001.

[126] Maarten van Dantzich, Daniel Robbins, Eric Horvitz, and Mary Czerwinski. Scope: Providing awareness of multiple notifications at a glance. In *Proceedings of the 6th International Working Conference on Advanced Visual Interfaces (AVI '02)*. ACM Press, 2002.

[127] Maarten van Dantzich, Daniel Robbins, Eric Horvitz, and Mary Czerwinski. Scope: Providing awareness of multiple notifications at a glance. In *Proceedings of the 6th International Working Conference on Advanced Visual Interfaces (AVI '02)*. ACM Press, 2002.

[128] Amy Voida, Wendy C. Newstetter, and Elizabeth D. Mynatt. When conventions collide: the tensions of instant messaging attributed. In *Proceeding of the CHI 02 Conference on Human Factors in Computing Systems*, pages 187–194. ACM Press, 2002.

[129] Shahtab Wahid, C. F. Allgood, C. M. Chewar, and D. Scott McCrickard. Entering the heart of design: Relationships for tracing claim evolution. In *Proceedings of the Sixteenth International Conference on Software Engineering and Knowledge Engineering (SEKE 2004)*, page 6 pgs, June 2004.

[130] Suzanne Watzman. Visual design principles for usable interfaces. In Julie A. Jacko and Andrew Sears, editors, *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, pages 263–285. Lawrence Erlbaum Associates, 2003.

[131] Timothy A. Weber, Arthur F. Kramer, and Gregory A. Miller. Selective processing of superimposed objects: An electrophysiological analysis of object-based attentional selection. In *ARL Advanced and Interactive Displays Symposium*, pages 33–38, February 1998.

[132] Mark Weiser and John Seely Brown. Designing calm technology. *PowerGrid Journal*, 1.01, July 1996.

[133] Steve Whittaker, Loren Terveen, and Bonnie A. Nardi. Let's stop pushing the envelope and start addressing it: A reference task agenda for HCI. *Human-Computer Interaction*, 15:75–106, 2000.

[134] C. D. Wickens, D. H. Merwin, and E. L. Lin. Implications of graphics enhancements for the visualization of scientific data: Dimensional integrality, stereopsis, motion and mesh. *Human Factors*, 36:44–61, 1994.

[135] Christopher D. Wickens and Justin G. Hollands. *Engineering Psychology and Human Performance*. Prentice Hall, Upper Saddle River, NJ, third edition, 2000.

[136] Stephanie Wilson and Peter Johnson. Bridging the generation gap: From work tools to user interface design. In Jean Vanderdonct, editor, *Computer-Aided Design of User Interfaces*, pages 77–94. Presses Universsitaires, 1996.

Appendix A

IRCspec Documentation, Testing Materials, and Results

A.1 Design Rationale Analysis

As part of a needs analysis for the development of the IRCspec tool, a design rationale analysis was conducted. For the first part of the study, eight papers published by experienced, professional researchers were selected. Papers citations for interfaces/studies included in Table A.1 are as follows: Scope [126], Sideshow [19], Unremarkable Computing [121], Kimera [78], ambientROOM [68], IM tensions [128], animation in online banners [14], interactive advertising [105].

For the second part of the study, design rationale provided in student project reports was analyzed. To preserve anonymity, we simply list the names of all students that generated material used in this study (included in Table A.2):

Ghada Abdelmoumin, Joshua Adell, Lyudmil Antonov, John Archie, Edwin Bachetti, Niteesh Bharara, Jamika Burge, Dillon Bussert, Lian Chen, Hyunjan Choi, Thomas Christ, Youngyun Chungback, Ben Congleton, Glenn Fink, Timothy Fuller, Paulette Goodman, Arshish Gupta, Alireza Hannani, Benjamin Hanrahan, Solomon Hardin, Glenn Hazelwood, Cody Henthome, Gregory Hightower, Brad Hunt, Andrew Jackson, Joseph Jezioro, Abhijeet Jhala, Aaron Kaluszka, Cris Kania, Theresa Klunk, Sujatha Krishnamoorthy, Sai Krishnan, Jedediah Lake, Vinay Lakhani, Will Lee, Dhruv Manek, Cyril Montabert, Benjamin Oravetz, Sourabh Pawar, Nicholas Polys, D. J. Shin, Timmon Wong, Lu Xiao, and Umur Yilmaz.

Table A.1: *Design rationale of experienced, professional developers and researchers, from published papers (continued on next page).*

Experienced, Professional Designers/Researchers

	Interface/ Study Focus	Authors' Affiliation	**Design/study rationale noted, related to…** Interruption	Reaction	Comprehension	Other
1	Scope (desktop application)	Microsoft	• "minimize distractions" • "manage higher rates of interruptions" • "minimal strain on cognitive resources" • "glanceable…easy to read and understand in a minimal amount of time" • "safeguarding a user's attention" • "empower users to stay focused on their primary task" • "unobtrusive display modalities"	• "handle alerts… status changes" • "decide what to attend to, and when" • "initiative primarily with the user" • "direct a user's attention to high urgency items" • helping the user "decide what to do next" • "low effort interactions, in order to decide whether to open the item in its native application"	• "remain aware" • "providing awareness of relevant information" • "stay aware of incoming notifications and pending tasks" • "monitoring incoming items" • "catching up on newly arrived items" • "implicit to-do list"	• adaptivity • prioritization • standardized/ normalized presentation • reconfigurability
2	Sideshow (desktop application)	Microsoft	• "without being overwhelmed or distracted" • "visually persistent in people's periphery" • variable distraction desired • "as visually calm as possible" • use of "screen real estate and [implications for] distraction"	• "launch point for easy drill down" • support user decision "to find out more information about a particular item" • "alert feature…to persist until clicked, or to fade away after a few seconds" • "easy for users to get highly detailed information"	• "help people stay aware of…a large number of information sources" • "increase the amount and value of information made accessible to the user" • "high-level summary of information in a small space" • "watch the majority of the important, dynamic information in one's world"	• versatile • extensible • customizable • relevant
3	General strategies for "Unremarkable computing"	Xerox-Europe	• "without having to eternally take pause and invent sequences of action anew" • "routines are invisible in user for those who are involved in them" • [aiming for] "tacit and calm…not overly 'dramatic' and do not command attention except when they need to" • "concerning perceptual psychology of 'peripheral sensory processing'" • [not just providing] "perceptually softer notifications" • "nothing inherent in the going off of the alarm that obliges her to treat it as a notable or remarkable event"	• [should leverage] "innumerable things we take for granted such that each ordinary enterprise can be undertaken unhesitatingly" • "used as resources for action…in ways that have a wealth of significance but have been made equally unremarkable" • "just enough to suffice as acknowledgement whilst..doing something else" • "unremarkably embedded into routines and augment action" • [relies on] "intelligibility of very specific courses of action"	• [providing] "a message, the import of which is only locally intelligible" • "just enough to [convey meaning]" • [relies on] "well understood mutual accountabilities" • "only intelligible at a very specific time of the day" • "knowing the routines of others can serve as a resource for an activity… discovered through noticing"	• acceptable for domestic (not office) settings
4	Kimera (augmented office environment)	Georgia Tech	• "peripheral interfaces that compliment existing focal work areas, and supporting the natural flow of work across these two setting" • "take advantage of people's uncanny ability to utilize peripheral information with comparatively little effort" • "off-load information into the physical environment" • "background awareness cues" • "causal inspection"	• "manage multiple working contexts" • "simultaneously organize, monitor, and manage multiple activities" • "supporting interaction with montages…integrat[ed] with background conceptual cues" • "jot a quick reminder" • "selecting a montage triggers a task switch" • "orchestrating their work between areas of focused activity…and peripheral areas of information that require minimal attention and interaction" • "links back to previous activities"	• "maintain awareness of background activities" • "support both awareness, and resumption, of background tasks" • "remind the user of past actions" • "designed to support… organizational activities" • "'keeping tabs on things'..harkens the need to constantly monitor multiple activities" • exposing relationships between information sources	• multi-desktop manager

5	ambientROOM prototypes (tangible user interfaces)	MIT	• "receiving various information from the 'periphery' without attending to it explicitly"
			• "smooth transition of users' focus of attention between background and foreground"
			• "delivery of computation should be transparent"
			• "visible and audible from many offices without being obtrusive"
			• "peripheral cues"
			• "employ both the background and foreground of users' attention"
			• "take advantage of this natural parallel background processing"
				• "providing handles for the seamless transition of the user's interaction between background and foreground information"
				• support transition for "displaying more detailed interactive graphical information"
					• "communicating information which is not [part of] the user's primary foreground task"
					• providing:
					  ○ "information from ambient sources"
					  ○ "an idea of the activities of colleagues"
						• poetic representation

6	Instant Messaging Tensions (related to specific tasks and goals)	Georgia Tech	• IM valued because "nearly synchronous but able to be attended to when opportune"
			• IM boring "unless she was already engaged elsewhere and could multitask" otherwise, maintain multiple threads
			• "participants frequently multitasked while instant messaging"
			• "did not feel they had to attend and respond right away"
			• "users stage workarounds to try to avoid giving a conversation their full attention"
				• "asynchronous nature…contributes to missed comments"
				• "visualizations supporting turn negotiation"
				• "users have felt socially compelled either to convey the illusion that instant messaging has their full attention or to offer justifications and preemptive repair tactics"
				• "manage their availability and to communicate context regarding their availability"
				• to support transient communication goals,
				  ○ "supporting short-hand or graffiti" or "text-fading or displays limited to the most recent statements"
					• for persistent communication goals,
					  ○ "integration of spell-checking r grammar checking"
					  ○ "maintaining a history"
					• "differentiate between multiple threads in conversation...denote state of that portion of the conversation"
					• providing "awareness cues" for "claiming a turn"
					• support for syntax comprehension: "allowing users to convey the state of their thoughts along with the text"
					• "explicit indicators of context"
						• written vs. verbal communication conventions

7	Animation in Online Banners	Wichita State Unv	• "users ignoring…even when presented as large and brightly colored" • "making the ads more noticeable"	• "main goal is to drive traffic to [other] websites" • "most extensively used measurement tool for ad effectiveness is the use of click-through rates"	• 'banner blindness' = users recalling existence of ads • "banner ads can create brand awareness, message association, purchase intent, and brand favorability" • "drive customers to purchase…without clicking on an ad" • "generate increase in purchase probability" • need to investigate "user awareness rather than click-through rates"	
8	Guidelines for Interactive Advertising Effectiveness	Microsoft & U Wash	• targeting "frequent and lengthy exposure to the product information" • "visited the ad at least once" • opportunity for high rates of exposure to the ad content…through roll over and clicking through" • "undisguised entertainment attractions designed to be fun" • "advertisements presented in different kinds of media will interact" • "exit button takes them back to the originating page of the sponsored website"	• "the actual purpose of many banners is to entice users to click through to the advertised company's website" • "anchors [should be] obviously 'clickable', to set up expectations for fun, and [use] highly directive language" • comparing "number of times they interacted with the ad content"	• "likely to result in superior memory for products" • interaction facilitates learning and memory for visual information" • "superior understanding and memory for ad content underlies product awareness" • "more prominent in subjects' memories" • "build ads which influence memory"	

Table A.2: *Design rationale from student design projects (continued on next two pages).*

Novice Designers (students)

C# Design Contest – Expedia Price Notification

	Interface	Design rationale noted, related to… Interruption	Reaction	Comprehension	Other
1	A	none	none	• "gives the user an easy outlook on the desired data (prices)"	
2	B	• "the design is simple"	none	none	
3	C	• "user can simply check this window periodically" • "the user can only minimize or close this window"	• "the user can go to Expedia.com to buy the ticket	• shows "if the prices are up or down" • "understanding price trends is fairly easy"	
4	D	• "the entire system at its fullest size is small enough not to distract the user. It will also not distract the user by taking away the current window's focus when updates are displayed" • "option of locking the slide in view" • "the slide will also appear every time the flight ticket information is updated"	• "a pop-up form [slide]…will appear if the user's cursor goes over the green tab" • "allows the user to identify whether it would be a good time to buy tickets, or if they would be better off waiting"	• "includes a graph of the price trend" • "can view the departure and destination airports, the departure and arrival times, and the price" • "all the tickets for an update are sorted according to price, and can be viewed by clicking the left and right mouse buttons on the ticket viewer control"	
5	E	none	none	• "prices are displayed next to the chosen location, which provide the user an intuitive grouping of information"	
6	F	• "quick and easy to read dials"	none	• "cheapest flight is shown to both destinations"	
7	G	none	none	• "basically displays the best prices" • "if the prices has decreased from the last time it will change the color of the price to green" • "if the price went high the price will be colored red"	
8	H	none	none	none	
9	I	none	none	• "easier visualization of the overall pricing distribution scheme" • "a max and min value are provided for each airport, so that users…gain a better understanding of the overall pricing trend"	

Notes:

Extracted from design rationale provided by contestants. Three prizes offered for "best usability": $400, $200, and $100. Design rationale instructions asked for: *"a description of the design model for your prototype detailing how you anticipate the interface will affect user interruption from ongoing tasks, reaction to Expedia information, and understanding of patterns and trends of ticket prices over time from the four locations. Also note any other important effects on users you anticipate."*

Basic design guidelines given were:

- The desktop display should be relatively small (not so small as to cause annoyance), and always visible (in some form).
- The display should not be annoying or offensive to view even after extended periods of use.
- The system use must be entirely intuitive for typical college-aged computer users; all features should be understandable without accessing a help file (tool tips are allowed).

Undergraduate Introduction to HCI (CS 3724) design project – The Scope redesign

Scope Alternatives

	Design/ rationale noted, related to…			
Interface	**Interruption**	**Reaction**	**Comprehension**	**Other**
A	• Claim: "Because the Scope fails to display necessary information, it also fails to be a glance-able display." • Proposal: "By using the **bulletin board metaphor**, we can include short "messages" or titles of each alert…. Before, if a user wanted to see what a task was, they must use tooltips and interact with the Scope. Now, the title of each alert will be presented to the user allowing the new Scope to be truly glancable." • BUT: "Using post-its make the screen more noisy."	• Claim: "By not providing enough details at first glance, the user must waste valuable time interacting with the system so that she might decide what action or inaction to take" • Proposal: "This will allow users to easily discern alerts with identical icons…. Users can now see the purpose of their alerts without guessing based on the limited data presented *via* icons."	• Claim: "The Scope simply fails to provide the user with enough information "at-a-glance" to be useful" • Proposal: "Text descriptions of messages significantly increase the scope's ability to convey information to a user…"	
C	• Claim: "With the current Scope design, the display takes up a forth of the screen. That is more distraction to the users, keeping the user away for the task. The Scope display needs to be changed in order to achieve the goal of reducing distractions to the user." • Proposal: A "**task bar** feature provides users with more available space on the screen and reduces the distractions to the users from the previous shape of Scope. Users can gain more space and have lesser distractions by using auto-hide command."	• Claim: "Many dots being close to each other, [a user] cannot figure out which ones are more important by just looking at them." • Proposal: "Sorting of messages according to the priorities from top to bottom reduce[s] complication of distinguishing the priorities of items…and easy to pick out items" • BUT: "users can miss urgency messages"	none	
E	• Claim: "The Scope quickly becomes cluttered and overwhelming when more applications are monitored." • Proposal: "a more usable shape for the alert notification program would be a **rectangle**… the user will be able to focus on just one rectangle easier than a segment of a circle, and will not be distracted by other segments as easily." • BUT: "slight loss of glance-ability"	none	• Claim: "becomes cluttered and hard to discern which blip on the radar belongs to what segment" • Proposal: "The rectangular design will also help user to find information about one particular type of event. A user's ability to perceive, interpret and make sense of which segment the event belongs to will be drastically improved due to the exponential decay of clarity as more applications are monitored with The Scope"	• Adding new segments easily
G	• Claim: "the scope can get in the way since it is circular shaped. Such an application would occlude other applications" • Proposal: "A sidebar solves the space problem and is friendly with other applications... Important information will now be available to a user with a simple glance at the screen."	• Claim: "the method of displaying high priority items does not necessarily take into account time." • Proposal: "the task icons will appear at or near the top of the interface at creation. As time elapses and the deadline for the task approaches, the icon progressively travels lower on the screen…Determining deadlines is easily accomplished."	• Claim: "We do not believe that these predefined categories are the best options for users and tasks will almost certainly not be evenly distributed, resulting in wasted space or uninterpretable data." • Proposal: "The inclusion of the **waterfall model**…[helps users] quickly perceive the meaning of the items on the interface and interpret the information, reducing the time it takes to make sense of the presented items"	
H	• Claim: "users possibly being frustrated by an application that was always on the screen constantly showing them pending tasks. This idea was found to cause unnecessary stress for the users, and it would also keep them from being able to focus on more important things on their computers." • Proposal: "we are redesigning Scope to have the capability of **running invisibly** in the background, and popping up when an item reaches a high priority level" • BUT: "it may interrupt the user at an inopportune time, causing frustration with the product"	• Claim: "dragging items from the scope view to delete them seemed too easy to do" • Proposal: "to avoid accidental deletions, we have changed the method of deleting items…a 'garbage can' at the bottom of the list"	• Claim: "people may not have used a radar or even have knowledge of how a radar works so the metaphor would be useless for them." • Proposal: "puts the tasks into a **list or a queue**… would get rid of the "incoming enemies" mentality, and hopefully replace it with a sense of being "on top of things."" • Claim: "the more important your tasks are the more cluttered in the display they become" • Proposal: "[having] the most important task would be at the top of the list and the least important would reside at the bottom of the list would eliminate the cluttering effect of approaching tasks. It is also obvious as to which tasks are of the utmost importance since they are on the top of the list."	• More classic

Notes:

Extracted from group reengineering proposals (phase 3 of the project). Phase instructions required students to focus on a single idea to guide redesign of Scope. This phase followed several weeks of effort in which Scope requirements analysis, activity, information, and interaction design deliverables were reconstructed (based on the published report).

Graduate Usability Engineering (CS 5714) design project – CNN notification system design

	Design rationale noted, related to…			
Interface	**Interruption**	**Reaction**	**Comprehension**	**Other**
A	• "(+) graphic rather than textual notification of news stories [because] less distracting" • "(-) graphical layout may take up more screen space than text" • "(-) too many colors may cause visual overload, [BUT we could] minimize the users' memory load [with a careful] color scheme…that could be distinguished out of the corner of one's eye" • "ensure that the interface was not annoying or distracting but highly usable with the interaction east and advantageous" • "a minimum of cognitive load is required to affect the [task] switch"	• [support] "processes of *content discovery*, *focus on news*, and *detailed news search* in tune with Shneiderman's mantra" • [graphics] "faster to interpret than text" • "identify the latest news clearly as it is on the top and is brighter than the older headlines" • "(-) but, headline text not directly visible without mouseover" • "(+) easier to locate individual stories" • "(-) grid may be too small for easy selection" • "less saturated colors look brighter and thus more important…imply increased importance...darker colors indicate story age" • "users to be able to easily switch from their primary task to a notification of interest" • "[to] reduce the cost of execution…determine a minimum acceptable size for the colored boxes and a maximum acceptable number of notifications that could be displayed" • "if the hyperlinked boxes are too small or too close, users may have difficulty selecting the content of interest" • "when a user notices a color or height change…they are one click away from an overview of that notification's content"	• "giving the user control to decide what detail of news is required" • [provide] "indication of when story is read" • "color distinguishes news stories…(+) groups stories by category and novelty" • "stacking bars to represent news headlines (+) [provides] immediate global idea of news content by category" • "bar height indicates number of stories (+) [as a] natural encoding of size to length" • "[glowing icon] indicates new news in that category"	
C	• [provide] "real time notifications" • BUT "(-) users may be annoyed by too many updates" • "pulsing animation… (+) shows feedback about state changes"	• "allow users to access premium site content" • "given easy and intuitive access to the link to support the activity of reading the complete article" • "(+) tooltips display the headline of the news without forcing the user to go to the entire article" • [turning a visited button grey] "keeps him from choosing the same headlines over again" • "bright colors…makes it easy to see [new notifications]" • "(-) icons placed next to each other may be difficult to click"	• "notifies users of news-related information in an informative way" • "a history of events can help the user keep abreast with all that has happened while he may have been away" • "(-) does not give more than the headline…even if the user wants a small abstract" • "colorful icons denote each of the categories making it easy to understand the kinds of news" • "(-) users may not know how recent headlines are if they miss [the pulsing animation]"	
G	• "monitor the latest information at a glance" • "notifying users when news are updated" • "displaying news icons and contents in the same window make the display more concise"	• "allow quick access to detailed information" • "recognizing news updates when they happened" • "access premium content without having to type a URL" • "icons might not be visible or available for access all the time" • "less user intervention is always desirable"	• [help users] "continue to follow up alerts after being away for a period of time" • "viewing concise and brief alert messages keeps users instantly informed on latest updates" • "might not be able to read alerts if display area is too small" • "displaying news categories and current time" • "different colors [darkened] distinguish the updated states"	• Customization

A.2 Algorithm, Screenshots, and Program Code

Screenshots and screen flow of the final version of IRCspec are provided on the pages that follow. IRCspec is a fully interactive Flash movie that can return parameters to a webpage or write a text file log. A standalone working version of IRCspec (requires Macromedia Flash Player 6) is available at http://research.cs.vt.edu/ns/IRCspec.exe. This version of IRCspec has also been integrated in the requirements analysis module of the LINK-UP system. Although not included in the final version, the initial underlying algorithm is provided below.

Initial IRCspec algorithm:

1. *Establish a base IRC with a small set of questions asked at the end of the process (allowing users to use the considerations from previous questions).*

2. *Infer the possible range of values that the base IRC can extend while staying within the user's direct specification (refer to this as the "base range").*

3. *Segment the dual-task characteristic questions into logical groups that would have clear and individual effects on IRC parameters, noting which parameter modifications should have larger weights than others. [In the initial flowchart depicted in Figure 5.3 these are shown as the green-shaded groups and referred to as a "MOD"].*

4. *Prepare a contingency table for each possible response combination, noting any constraining implication for an I, R, and C value that should result if that combination is invoked as a rule (e.g. "$I > .75$" could be set as a rule if a user specifies that the notifications delivered by a particular system are almost always more important than the primary task).*

5. *Aggregate the set of rules from all question groups (taking into account any parameter weighting) to specify the most constraining modification conditions for each parameter.*

6. *Test the base range of IRC values (each parameter separately) against the modification conditions to ensure consistency. If no part of the base range falls within the constraints, reestablish the user's base IRC with a different set of question and retest. If there is still disagreement, repeat the question group causing disagreement and recheck (eventually flagging result if disagreement continues).*

7. *If the constraints divide the base range, set the final IRC value as the end of the acceptable portion of the base range that overlaps the constraint.*

8. *If the entire base range falls within the constraints, identify the direction of the constraint (i.e. minimizing or maximizing a parameter) and take the corresponding end of the base range as the final IRC value.*

This process, demonstrated in Figure A.1, will yield three values, one for each of the critical parameters that describe the IRC design model.

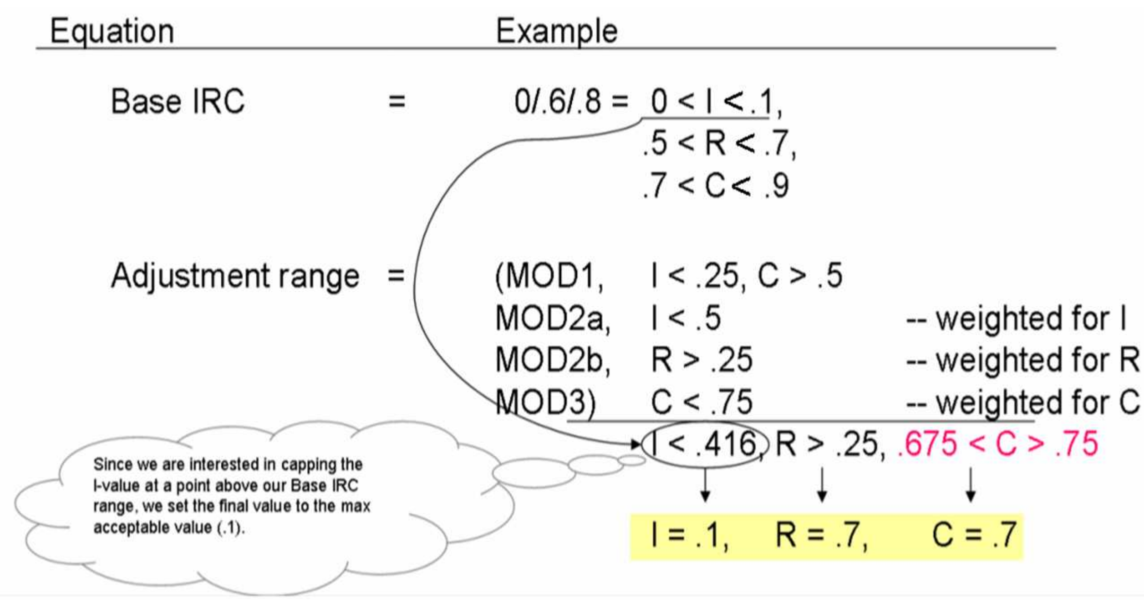
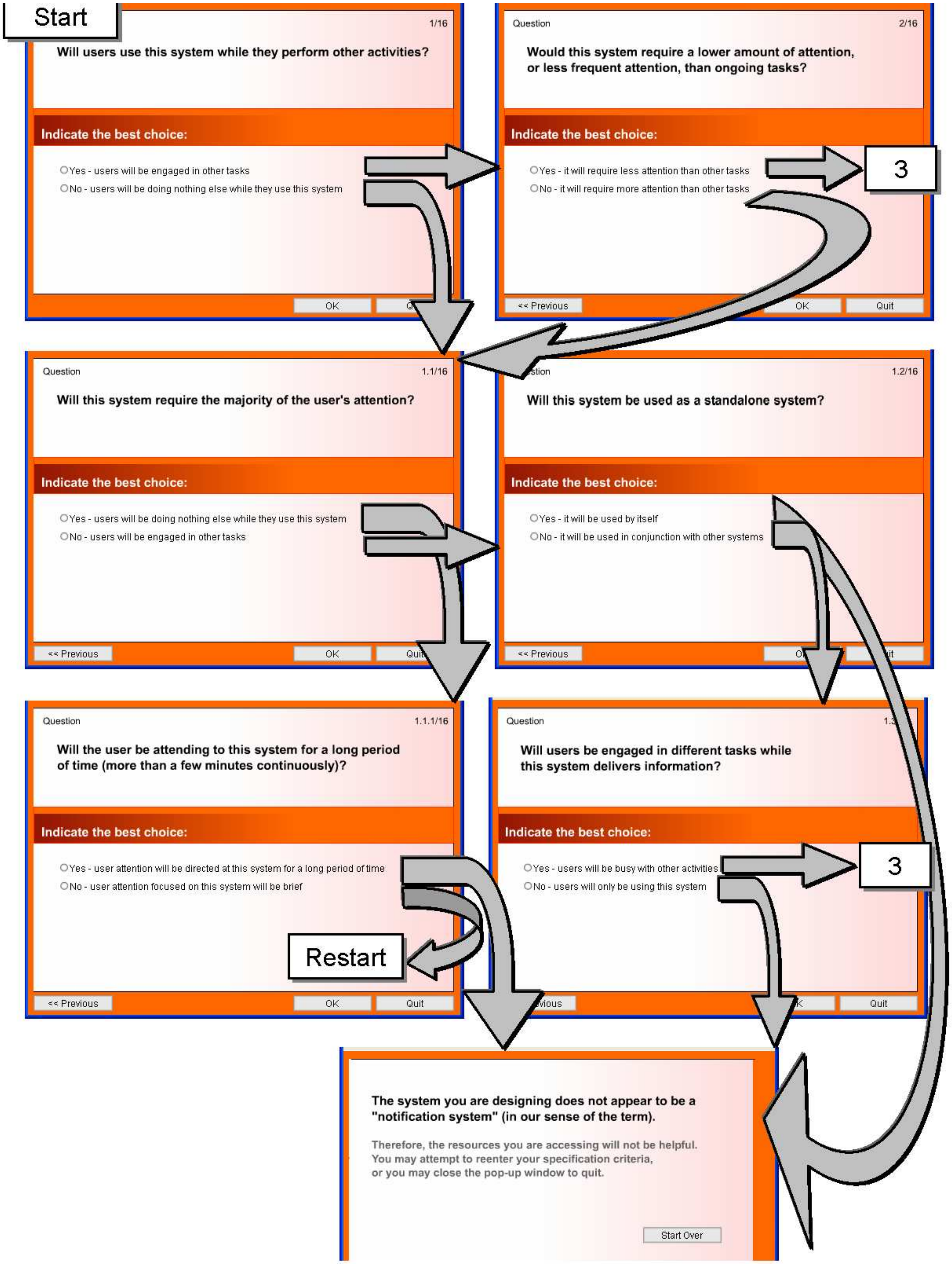
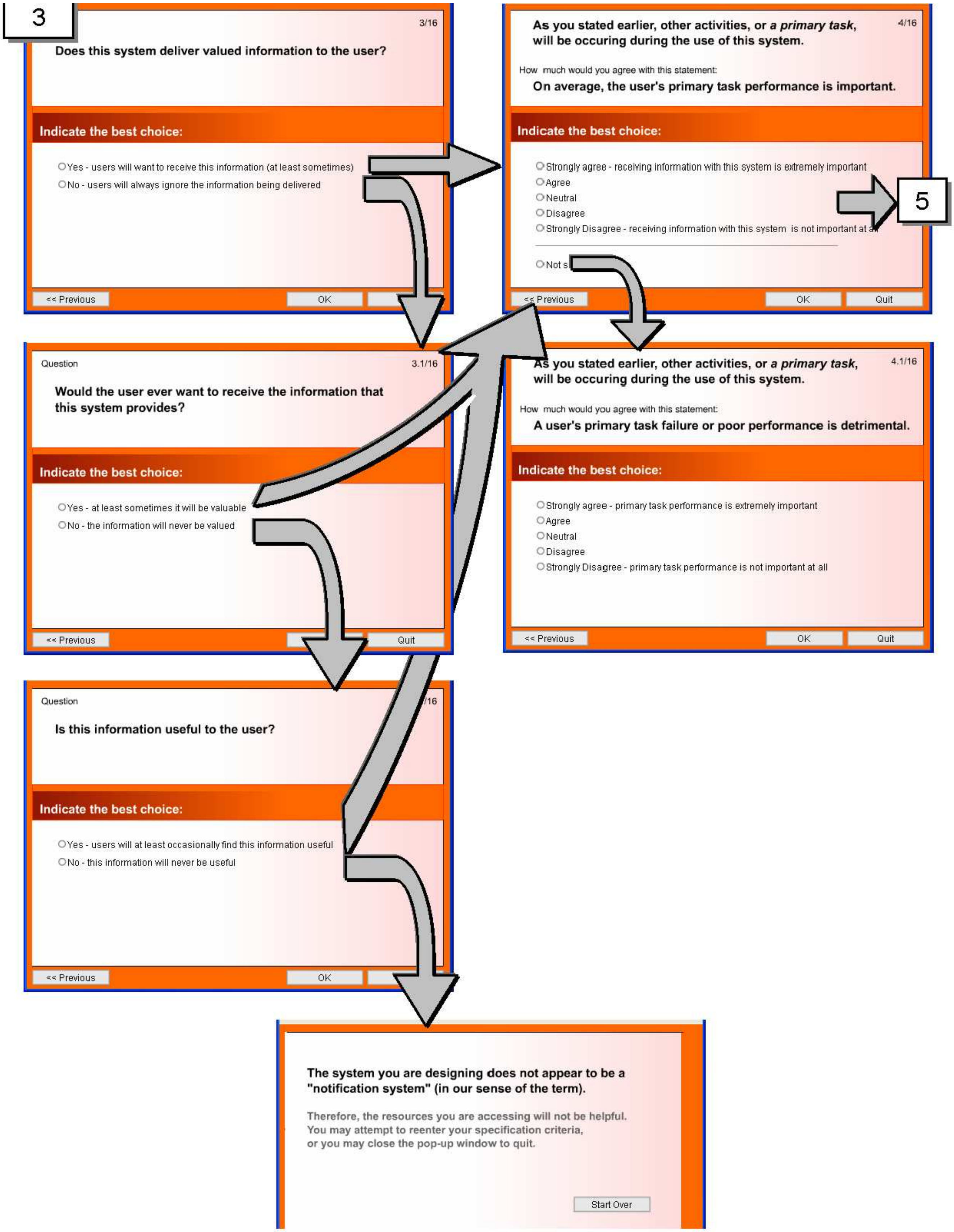
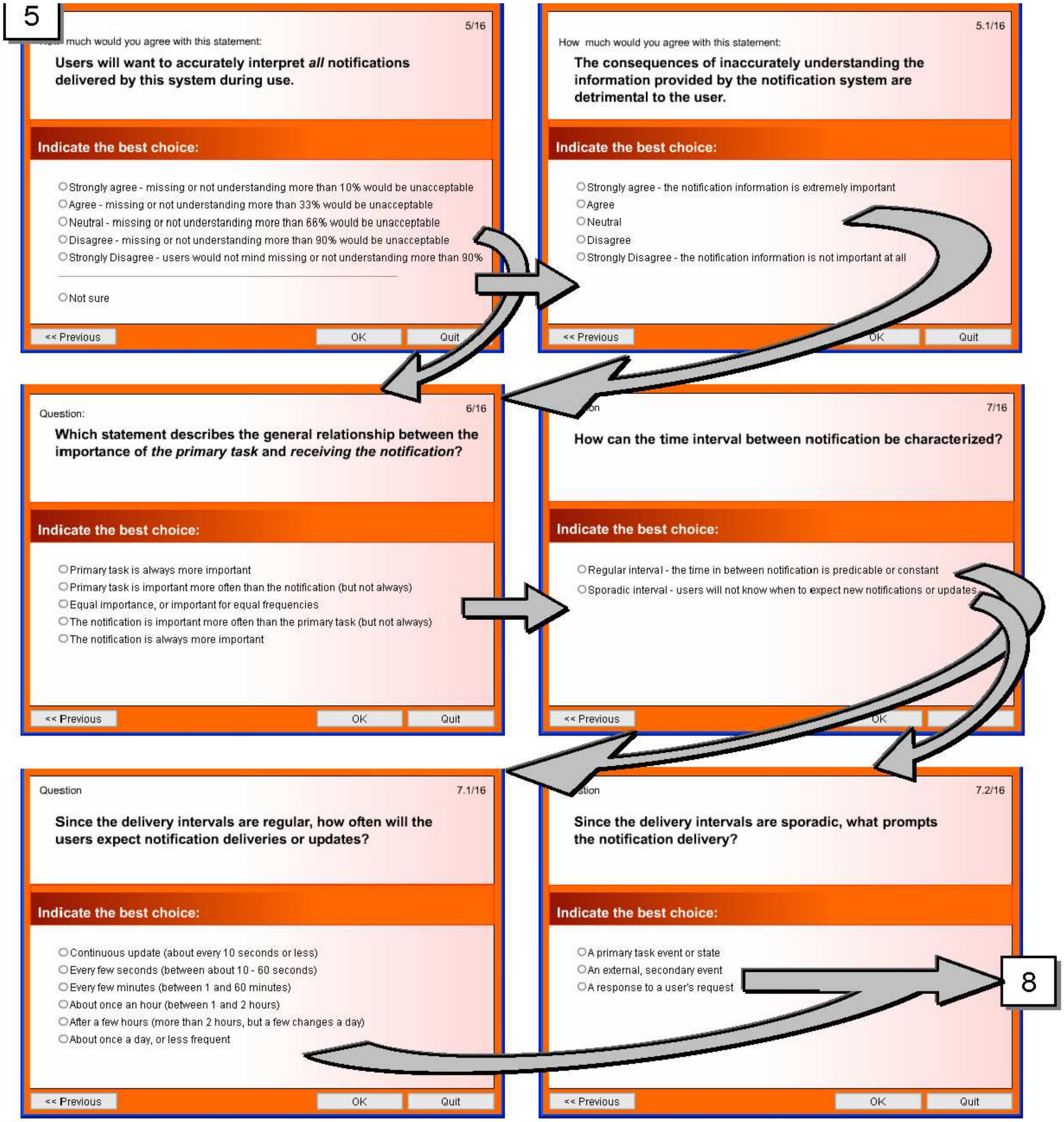
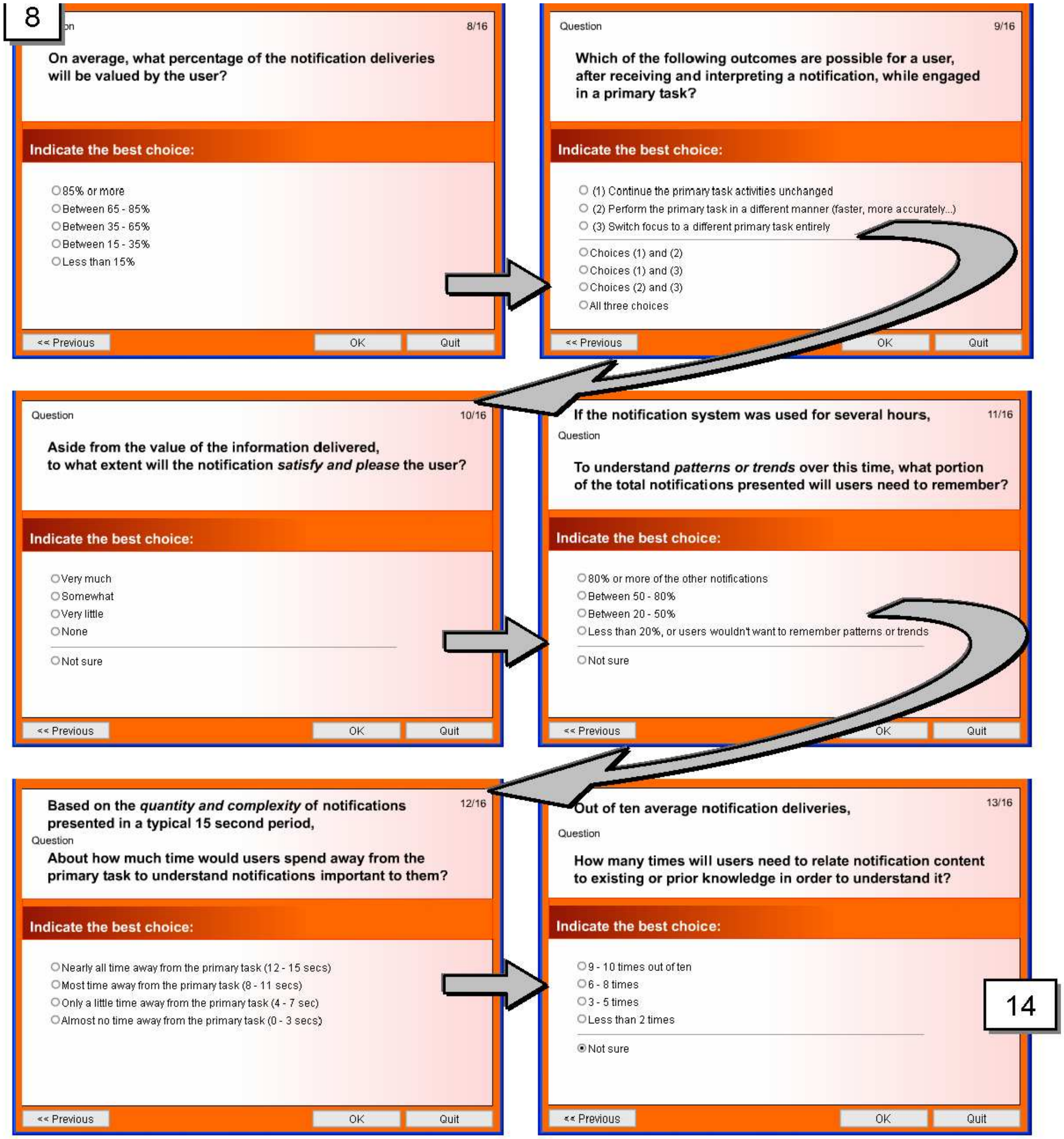


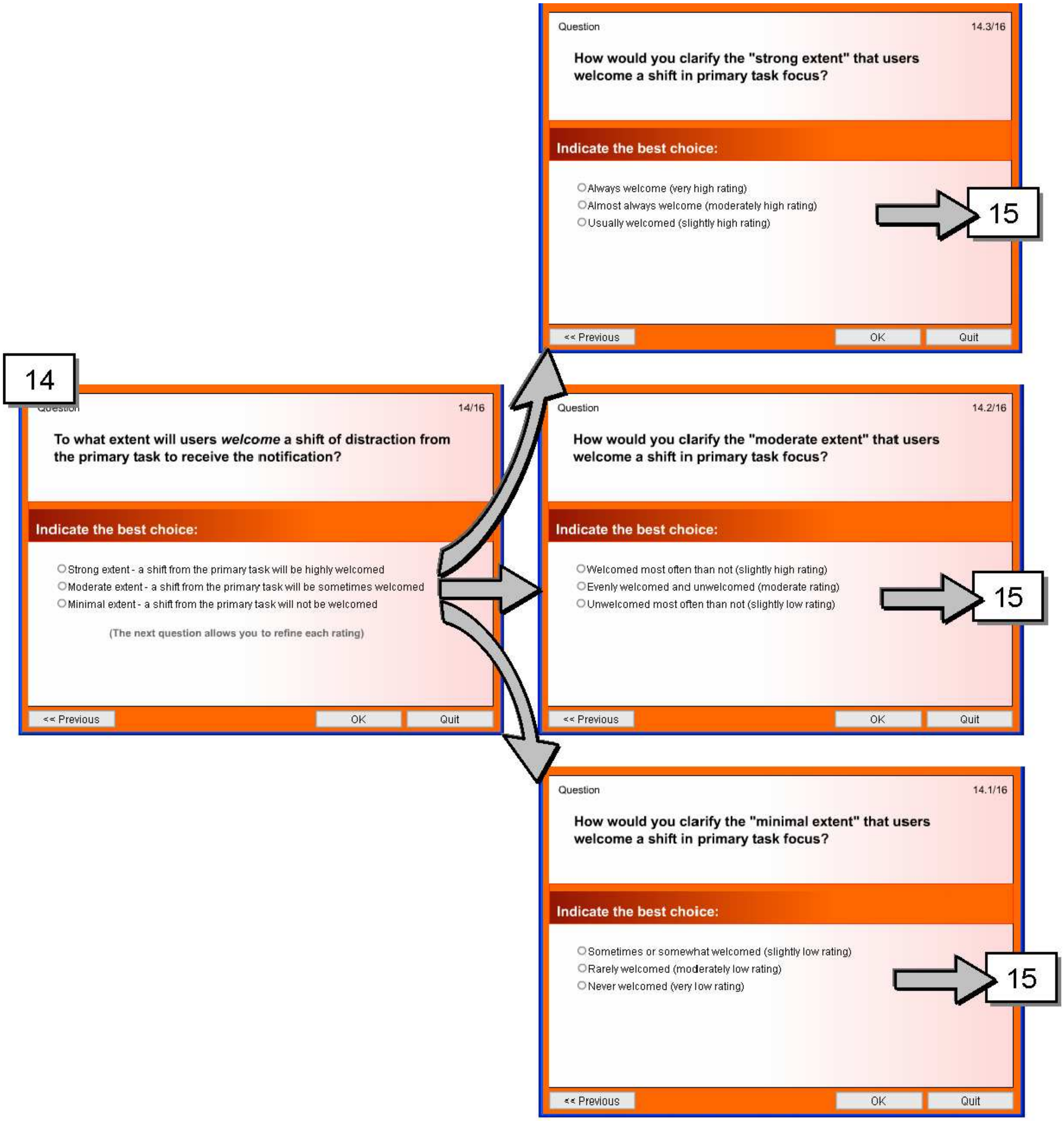
Figure A.1: Example design model IRC calculation process.


Start
1/16
Will users use this system while they perform other activities?
Indicate the best choice:
Yes - users will be engaged in other tasks
No - users will be doing nothing else while they use this system
OK
Question
2/16
Would this system require a lower amount of attention, or less frequent attention, than ongoing tasks?
Indicate the best choice:
Yes - it will require less attention than other tasks
No - it will require more attention than other tasks
3
<< Previous
OK
Quit
Question
1.1/16
Will this system require the majority of the user's attention?
Indicate the best choice:
Yes - users will be doing nothing else while they use this system
No - users will be engaged in other tasks
<< Previous
OK
1.2/16
Will this system be used as a standalone system?
Indicate the best choice:
Yes - it will be used by itself
No - it will be used in conjunction with other systems
<< Previous
Question
1.1.1/16
Will the user be attending to this system for a long period of time (more than a few minutes continuously)?
Indicate the best choice:
Yes - user attention will be directed at this system for a long period of time
No - user attention focused on this system will be brief
Restart
<< Previous
OK
Quit
Question
Will users be engaged in different tasks while this system delivers information?
Indicate the best choice:
Yes - users will be busy with other activities
No - users will only be using this system
3
Quit
The system you are designing does not appear to be a "notification system" (in our sense of the term).
Therefore, the resources you are accessing will not be helpful.
You may attempt to reenter your specification criteria,
or you may close the pop-up window to quit.
Start Over

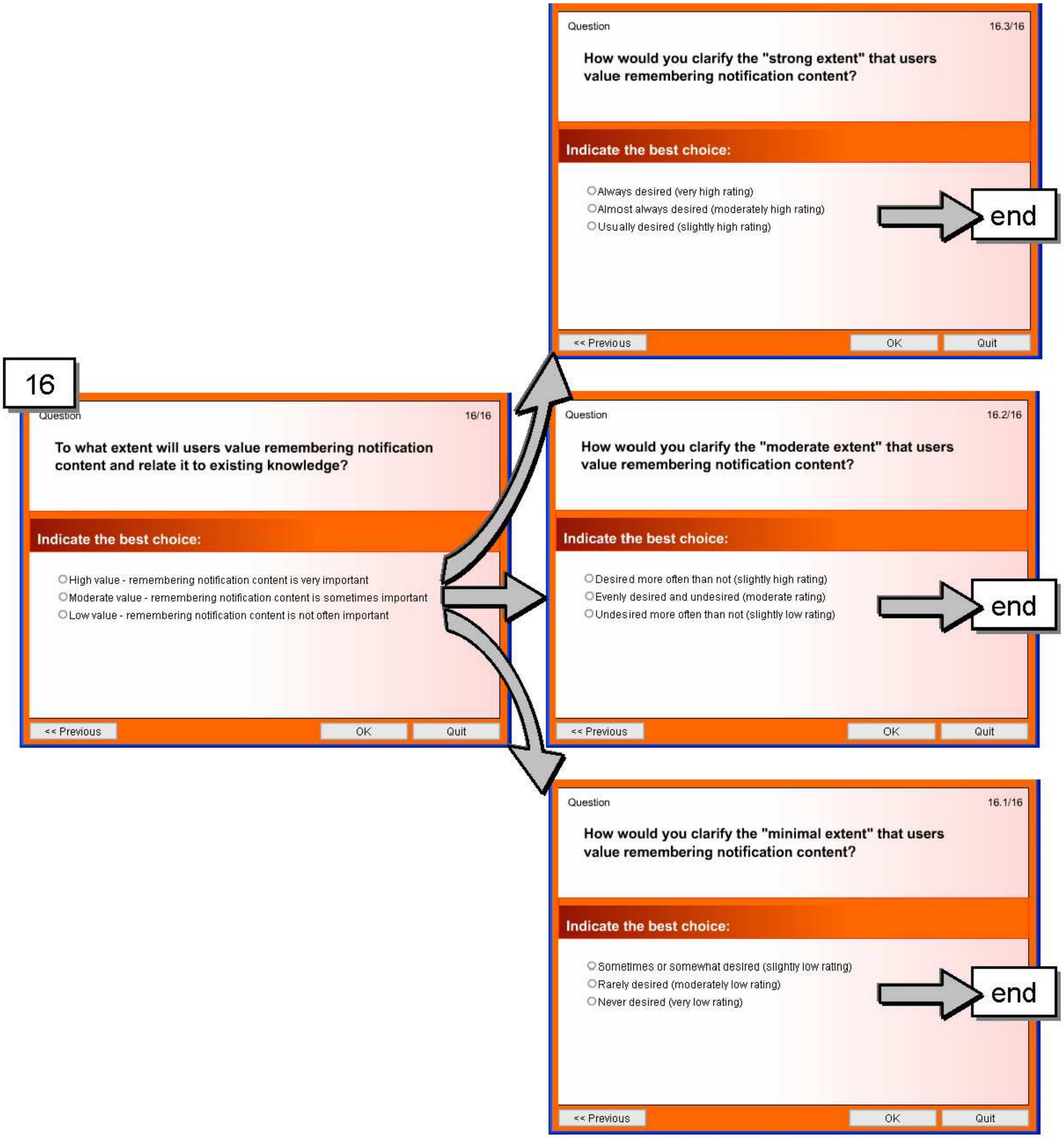

3
3/16
Does this system deliver valued information to the user?
Indicate the best choice:
Yes - users will want to receive this information (at least sometimes)
No - users will always ignore the information being delivered
<< Previous
OK
As you stated earlier, other activities, or a *primary task*, will be occuring during the use of this system.
4/16
How much would you agree with this statement:
On average, the user's primary task performance is important.
Indicate the best choice:
Strongly agree - receiving information with this system is extremely important
Agree
Neutral
Disagree
Strongly Disagree - receiving information with this system is not important at
5
<< Previous
OK
Quit
Question
3.1/16
Would the user ever want to receive the information that this system provides?
Indicate the best choice:
Yes - at least sometimes it will be valuable
No - the information will never be valued
<< Previous
Quit
As you stated earlier, other activities, or a *primary task*, will be occuring during the use of this system.
4.1/16
How much would you agree with this statement:
A user's primary task failure or poor performance is detrimental.
Indicate the best choice:
Strongly agree - primary task performance is extremely important
Agree
Neutral
Disagree
Strongly Disagree - primary task performance is not important at all
<< Previous
OK
Quit
Question
Is this information useful to the user?
Indicate the best choice:
Yes - users will at least occasionally find this information useful
No - this information will never be useful
<< Previous
OK
The system you are designing does not appear to be a "notification system" (in our sense of the term).
Therefore, the resources you are accessing will not be helpful.
You may attempt to reenter your specification criteria,
or you may close the pop-up window to quit.
Start Over

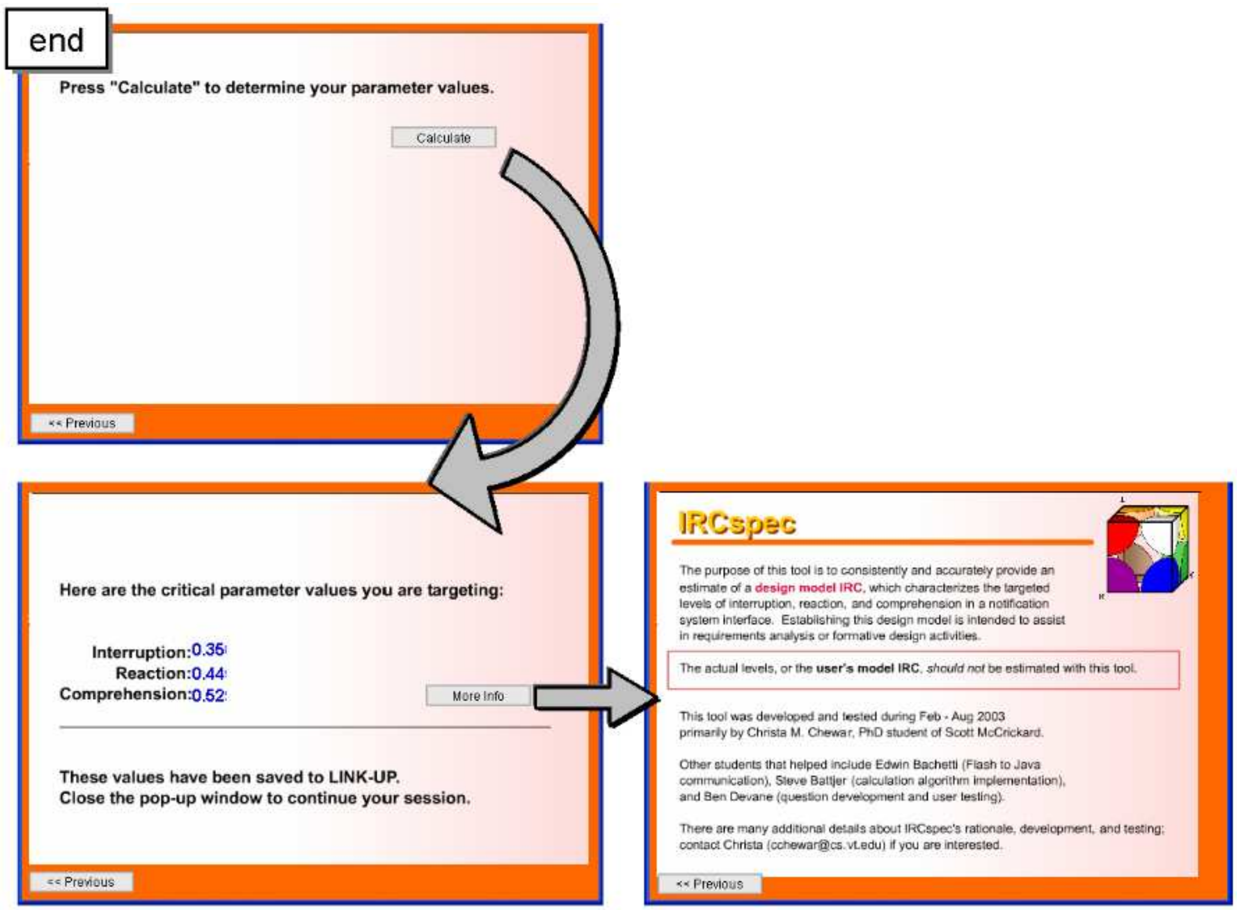

5
much would you agree with this statement:
5/16
Users will want to accurately interpret *all* notifications delivered by this system during use.
Indicate the best choice:
Strongly agree - missing or not understanding more than 10% would be unacceptable
Agree - missing or not understanding more than 33% would be unacceptable
Neutral - missing or not understanding more than 66% would be unacceptable
Disagree - missing or not understanding more than 90% would be unacceptable
Strongly Disagree - users would not mind missing or not understanding more than 90%
Not sure
<< Previous
OK
Quit
How much would you agree with this statement:
5.1/16
The consequences of inaccurately understanding the information provided by the notification system are detrimental to the user.
Indicate the best choice:
Strongly agree - the notification information is extremely important
Agree
Neutral
Disagree
Strongly Disagree - the notification information is not important at all
<< Previous
OK
Quit
Question:
6/16
Which statement describes the general relationship between the importance of *the primary task* and *receiving the notification*?
Indicate the best choice:
Primary task is always more important
Primary task is important more often than the notification (but not always)
Equal importance, or important for equal frequencies
The notification is important more often than the primary task (but not always)
The notification is always more important
<< Previous
OK
Quit
7/16
How can the time interval between notification be characterized?
Indicate the best choice:
Regular interval - the time in between notification is predicable or constant
Sporadic interval - users will not know when to expect new notifications or updates
<< Previous
OK
Question
7.1/16
Since the delivery intervals are regular, how often will the users expect notification deliveries or updates?
Indicate the best choice:
Continuous update (about every 10 seconds or less)
Every few seconds (between about 10 - 60 seconds)
Every few minutes (between 1 and 60 minutes)
About once an hour (between 1 and 2 hours)
After a few hours (more than 2 hours, but a few changes a day)
About once a day, or less frequent
<< Previous
OK
Quit
7.2/16
Since the delivery intervals are sporadic, what prompts the notification delivery?
Indicate the best choice:
A primary task event or state
An external, secondary event
A response to a user's request
8
<< Previous
OK
Quit


8
8/16
On average, what percentage of the notification deliveries will be valued by the user?
Indicate the best choice:
85% or more
Between 65 - 85%
Between 35 - 65%
Between 15 - 35%
Less than 15%
<< Previous
OK
Quit
Question
9/16
Which of the following outcomes are possible for a user, after receiving and interpreting a notification, while engaged in a primary task?
Indicate the best choice:
(1) Continue the primary task activities unchanged
(2) Perform the primary task in a different manner (faster, more accurately...)
(3) Switch focus to a different primary task entirely
Choices (1) and (2)
Choices (1) and (3)
Choices (2) and (3)
All three choices
<< Previous
OK
Quit
Question
10/16
Aside from the value of the information delivered, to what extent will the notification satisfy and please the user?
Indicate the best choice:
Very much
Somewhat
Very little
None
Not sure
<< Previous
OK
Quit
If the notification system was used for several hours,
11/16
Question
To understand patterns or trends over this time, what portion of the total notifications presented will users need to remember?
Indicate the best choice:
80% or more of the other notifications
Between 50 - 80%
Between 20 - 50%
Less than 20%, or users wouldn't want to remember patterns or trends
Not sure
<< Previous
OK
Quit
Based on the quantity and complexity of notifications presented in a typical 15 second period,
12/16
Question
About how much time would users spend away from the primary task to understand notifications important to them?
Indicate the best choice:
Nearly all time away from the primary task (12 - 15 secs)
Most time away from the primary task (8 - 11 secs)
Only a little time away from the primary task (4 - 7 sec)
Almost no time away from the primary task (0 - 3 secs)
<< Previous
OK
Quit
Out of ten average notification deliveries,
13/16
Question
How many times will users need to relate notification content to existing or prior knowledge in order to understand it?
Indicate the best choice:
9 - 10 times out of ten
6 - 8 times
3 - 5 times
Less than 2 times
Not sure
14
<< Previous
OK
Quit


14
Question
14/16
To what extent will users welcome a shift of distraction from the primary task to receive the notification?
Indicate the best choice:
Strong extent - a shift from the primary task will be highly welcomed
Moderate extent - a shift from the primary task will be sometimes welcomed
Minimal extent - a shift from the primary task will not be welcomed
(The next question allows you to refine each rating)
<< Previous
OK
Quit
Question
14.3/16
How would you clarify the "strong extent" that users welcome a shift in primary task focus?
Indicate the best choice:
Always welcome (very high rating)
Almost always welcome (moderately high rating)
Usually welcomed (slightly high rating)
15
<< Previous
OK
Quit
Question
14.2/16
How would you clarify the "moderate extent" that users welcome a shift in primary task focus?
Indicate the best choice:
Welcomed most often than not (slightly high rating)
Evenly welcomed and unwelcomed (moderate rating)
Unwelcomed most often than not (slightly low rating)
15
<< Previous
OK
Quit
Question
14.1/16
How would you clarify the "minimal extent" that users welcome a shift in primary task focus?
Indicate the best choice:
Sometimes or somewhat welcomed (slightly low rating)
Rarely welcomed (moderately low rating)
Never welcomed (very low rating)
15
<< Previous
OK
Quit


15
15/16
What value will users place on reacting to the content of the notification?
Indicate the best choice:
High value - proper and/or prompt reaction to the notification is very important
Moderate value - reaction to the notification is sometimes important
Minimal value - reaction is usually not important
(The next question allows you to refine each rating)
<< Previous
OK
Quit
Question
15.3/16
How would you clarify the "high value" that users place on reacting to the notification?
Indicate the best choice:
Always valued (very high rating)
Almost always valued (moderately high rating)
Usually valued (slightly high rating)
16
<< Previous
OK
Quit
Question
15.2/16
How would you clarify the "moderate value" that users place on reacting to the notification?
Indicate the best choice:
Valued more often than not (slightly high rating)
Evenly valued and not valued (moderate rating)
Not valued most often (slightly low rating)
16
<< Previous
OK
Quit
Question
15.1/16
How would you clarify the "minimal value" that users place on reacting to the notification?
Indicate the best choice:
Sometimes or somewhat valued (slightly low rating)
Rarely valued (moderately low rating)
Never valued (very low rating)
16
<< Previous
OK
Quit


16
Question
16/16
To what extent will users value remembering notification content and relate it to existing knowledge?
Indicate the best choice:
High value - remembering notification content is very important
Moderate value - remembering notification content is sometimes important
Low value - remembering notification content is not often important
<< Previous
OK
Quit
Question
16.3/16
How would you clarify the "strong extent" that users value remembering notification content?
Indicate the best choice:
Always desired (very high rating)
Almost always desired (moderately high rating)
Usually desired (slightly high rating)
end
Question
16.2/16
How would you clarify the "moderate extent" that users value remembering notification content?
Indicate the best choice:
Desired more often than not (slightly high rating)
Evenly desired and undesired (moderate rating)
Undesired more often than not (slightly low rating)
end
Question
16.1/16
How would you clarify the "minimal extent" that users value remembering notification content?
Indicate the best choice:
Sometimes or somewhat desired (slightly low rating)
Rarely desired (moderately low rating)
Never desired (very low rating)
end


end
Press "Calculate" to determine your parameter values.
Calculate
<< Previous
Here are the critical parameter values you are targeting:
Interruption:0.36
Reaction:0.44
Comprehension:0.52
More Info
These values have been saved to LINK-UP.
Close the pop-up window to continue your session.
<< Previous
IRCspec
The purpose of this tool is to consistently and accurately provide an estimate of a design model IRC, which characterizes the targeted levels of interruption, reaction, and comprehension in a notification system interface. Establishing this design model is intended to assist in requirements analysis or formative design activities.
The actual levels, or the user's model IRC, should not be estimated with this tool.
This tool was developed and tested during Feb - Aug 2003 primarily by Christa M. Chewar, PhD student of Scott McCrickard.
Other students that helped include Edwin Bachetti (Flash to Java communication), Steve Battjer (calculation algorithm implementation), and Ben Devane (question development and user testing).
There are many additional details about IRCspec's rationale, development, and testing; contact Christa (cchewar@cs.vt.edu) if you are interested.
<< Previous

```
function onClick(btn) {
		_global.screens.push("Calculate");
	if (btn == Calculate) {
		init_vars();
		main();
		trace(FinalI);
		_global.FinalI = FinalI;
		trace(FinalR);
		_global.FinalR = FinalR;
		trace(FinalC);
		_global.FinalC = FinalC;
		fscommand("send", FinalI + "," + FinalR + ","
+ FinalC);
		gotoAndStop("showIRC");
	} else if (btn == Previous) {
		_global.screens.pop();
		_global.screens.pop();
			trace(_global.screens);
		previousScreen = _global.screens.pop();
		gotoAndStop(previousScreen);
	}
}

function init_vars() {
	COI = 0;
	sust = 0;
	time = 0;
	hits = 0;
	percept = 0;
	fut = 0;
	comp = 0;
	utility = 0;
}

function main() {
	_global.IGT;
	_global.ILT;
	_global.RGT;
	_global.RLT;
	_global.CGT;
	_global.CLT;

	_global.BaseI;
	_global.BaseR;
	_global.BaseC;
	_global.IBot;
	_global.ITop;
	_global.RBot;
	_global.RTop;
	_global.CBot;
	_global.CTop;
	_global.FinalI;
	_global.FinalR;
	_global.FinalC;

	_global.numResults =0;
	//var results[20];   - will make this a global
variable
	var spot;


	IGT = ILT = RGT = RLT = CGT = CLT = BaseI = BaseR =
BaseC = IBot = ITop = RBot = RTop = CBot = CTop = FinalI =
FinalR = FinalC = 0;
	ILT = RLT = CLT = 1;

	if( q4 == 0) {
		q4 = q4x1;
	}
	if( q5 == 0) {
		q5 = q5x1;
	}

	_global.results = new
Array(q4,q5,q6,q7,q7b,q8,q9,q10,q11,q12,q13,q14,q14b,q15,q1
5b,q16,q16b);
	trace(results);

	//getResults(fin, results, numResults);  - don't
need

	spot = newMod1();
	spot = newMod2a(spot);
	spot = newMod3(spot);
	equation();

	correct();
	getBase(spot);
	getPos();
	getFinal();

}


function newMod1() {
	var q4, q5, q6, diff_q4q5;
	var spot =0;

	q4 = results[spot];
	trace("q4= " +q4);
	spot++;
	q5 = results[spot];
	trace("q5= " +q5);
	spot++;
	q6 = results[spot];
	trace("q6= " +q6);
	spot++;

	if( q5 == 5) {
		hits = 1;
	}
	if( q5 == 4) {
		hits = .75;
	}
	if( q5 == 3) {
		hits = .5;
	}
	if( q5 == 2) {
		hits = .25;
	}
	if( q5 == 1) {
		hits = .0001;
	}
	if( q6 == 5) {
		COI = 1;
	}
	if( q6 == 4) {
		COI = .75;
	}
	if( q6 == 3) {
		COI = .5;
	}
	if( q6 == 2) {
		COI = .25;
	}
	if( q6 == 1) {
		COI = .0001;
	}

	diff_q4q5 = q4 - q5;

	if ( diff_q4q5 == 4) {
		utility = 0;
	}
	if ( diff_q4q5 == 3) {
		utility = .15;
	}
	if ( diff_q4q5 == 2) {
		utility = .30;
	}
	if ( diff_q4q5 == 1) {
		utility = .45;
	}
	if ( diff_q4q5 == 0) {
		utility = .6;
	}
	if ( diff_q4q5 == -1) {
		utility = .7;
	}
	if ( diff_q4q5 == -2) {
		utility = .8;
	}
	if ( diff_q4q5 == -3) {
		utility = .9;
```

```
        }
        if ( diff_q4q5 == -4) {
                utility = 1;
        }

        return spot;
}

function newMod2a(spot) {
        var q7, q8, q9;
        if(results[spot] == 2) {
                spot++;
                q7 = results[spot];
                        trace("q7(2)= " +q7);
                spot++;
                q8 = results[spot];
                trace("q8= " +q8);
                spot++;
                q9 = results[spot];
                trace("q9= " +q9);
                spot++;
                if(q7 == 6) {
                        time = 1;
                }
                if(q7 == 5) {
                        time = .8;
                }
                if(q7 == 4) {
                        time = .6;
                }
                if(q7 == 3) {
                        time = .4;
                }
                if(q7 == 2) {
                        time = .2;
                }
                if(q7 == 1) {
                        time = .0001;
                }
        }
        else if(results[spot] == 1) {
                spot++;
                q7 = results[spot];
                trace("q7(1)= " +q7);
                spot++;
                q8 = results[spot];
                trace("q8= " +q8);
                spot++;
                q9 = results[spot];
                trace("q9= " +q9);
                spot++;

                if(q7 == 3) {
                        time = .75;
                }
                if(q7 == 2) {
                        time = .5;
                }
                if(q7 == 1) {
                        time = .25;
                }
        }
        if(q8 == 5) {
                percept = 1;
        }
        if(q8 == 4) {
                percept = .75;
        }
        if(q8 == 3) {
                percept = .5;
        }
        if(q8 == 2) {
                percept = .25;
        }
        if(q8 == 1) {
                percept = .0001;
        }
        if (q9 == 7) {
                comp = .25;
                if (utility < .1) {
                        comp = 0;
                }
        }
        if(q9 == 6) {
                if (hits < 1) {
                        hits = hits +.25;
                }
                if (percept < 1) {
                        percept = percept +.25;
                }
        }
        if(q9 == 5) {
                sust = -.25;
                if (time == .8) {
                        time = 1;
                }
                if (time < 1) {
                        time = time +.25;
                }
        }
        if(q9 == 4) {
                comp = .25;
                if (hits < 1) {
                        hits = hits +.25;
                }
                if (percept < 1) {
                        percept = percept +.25;
                }
        }
        if(q9 == 3) {
                comp = .25;
                if (time == .8) {
                        time = 1;
                }
                if (time < 1) {
                        time = time +.25;
                }
        }
        if(q9 == 2) {
                if (hits < 1) {
                        hits = hits +.25;
                }
                if (percept < 1) {
                        percept = percept +.25;
                }
                if (time == .8) {
                        time = 1;
                }
                if (time < 1) {
                        time = time +.25;
                }
        }
        if(q9 == 1) {
                comp = .25;
                if (hits < 1) {
                        hits = hits +.25;
                }
                if (percept < 1) {
                        percept = percept +.25;
                }
                if (time == .8) {
                        time = 1;
                }
                if (time < 1) {
                        time = time +.25;
                }
        }

        return spot;
}

function newMod3(spot) {
        var q11, q12, q13;
        spot++;
        q11 = results[spot];
        trace("q11= " +q11);
        spot++;
        q12 = results[spot];
        trace("q12= " +q12);
        spot++;
        q13 = results[spot];
        trace("q13= " +q13);
        spot++;
```

```
	if(q11 == 5) {
		fut = 1;
	}
	if(q11 == 4) {
		fut = .66;
	}
	if(q11 == 3) {
		fut = .33;
	}
	if(q11 == 2) {
		fut = .0001;
	}
	if(q12 == 4) {
		sust = .0001;
	}
	if(q12 == 3) {
		sust = sust + .5;
	}
	if(q12 == 2) {
		sust = sust + .75;
	}
	if(q12 == 1) {
		sust = sust + 1;
	}
	if(q13 == 5) {
		comp = 1;
	}
	if(q13 == 4) {
		comp = comp + .66;
	}
	if(q13 == 3) {
		comp = comp + .33;
	}
	if(q13 == 2) {
		comp = comp + .0001;
	}
	return spot;
}

function equation() {
	trace("sust =" + sust);
	trace("COI =" + COI);
	trace("hits =" + hits);
	trace("time =" + time);
	trace("percept =" + percept);
	trace("comp =" + comp);
	trace("fut =" + fut);

	var tripleCOI = 3 * COI;
	var one_over = 1 / tripleCOI;
	var i_exp = Math.pow(sust,tripleCOI);
	var r_exp = Math.pow((time * hits), one_over);
	trace("r_exp =" + r_exp);

	var i_value = 1 - i_exp;

	var r_1 = r_exp / 2;
	var r_2 = (.5 + COI) * hits / 3;
	var r_value = r_1 + r_2;
	var c_value = fut + ((1 - fut)*(percept + (2* comp)
- (comp * percept)))/3;
	trace("I value =" + i_value);
	trace("R value =" + r_value);
	trace("C value =" + c_value);


	if (utility > r_value && utility > c_value) {
		var diff_cutil = utility - c_value;
		var diff_rutil = utility - r_value;
		if (c_value == r_value) {
			c_value = c_value + diff_cutil;
			if (c_value > 1) {
				c_value = 1;
			}
			r_value = r_value + diff_rutil;
			if (r_value > 1) {
				r_value = 1;
			}
		}
		if (r_value > c_value) {
			r_value = r_value + diff_rutil;
			if (r_value > 1) {
				r_value = 1;
			}
		}
		if (c_value > r_value) {
			c_value = c_value + diff_cutil;
			if (c_value > 1) {
				c_value = 1;
			}
		}
	}

	ILT = i_value;
	IGT = i_value;
	RLT = r_value;
	RGT = r_value;
	CLT = c_value;
	CGT = c_value;

}


function correct() {
	var x;
	if(ILT == 0 && IGT != 0) {
		ILT = 1;
	}
	if(RLT == 0 && RGT != 0) {
		RLT = 1;
	}
	if(CLT == 0 && CGT != 0) {
		CLT = 1;
	}
	if(IGT > ILT) {
		x = IGT;
		IGT = ILT;
		ILT = x;
	}
	if(RGT > RLT) {
		x = RGT;
		RGT = RLT;
		RLT = x;
	}
	if(CGT > CLT) {
		x = CGT;
		CGT = CLT;
		CLT = x;
	}

}


function getBase(spot) {

	var q14, q15, q16;


	q14 = results[spot];
	spot++;
	trace("q14: " + q14);
	if(q14 == 3) {
		q14 = results[spot];
		spot++;
		if(q14 == 3) {
			BaseI = 1;
		}
		else if(q14 == 2) {
			BaseI = .8;
		}
		else {
			BaseI = .6;
		}
	}
	else if(q14 == 1) {
		q14 = results[spot];
		spot++;
		if(q14 == 3) {
			BaseI = .4;
		}
		else if(q14 == 2) {
			BaseI = .2;
		}
		else {
```

```
                BaseI = 0;
        }
}
else if(q14 == 2) {
        q14 = results[spot];
        spot++;
        if(q14 == 3) {
                BaseI = .6;
        }
        else if(q14 == 2) {
                BaseI = .5;
        }
        else {
                BaseI = .4;
        }
}
else {
        trace("ERROR IN Q14 ANALYSIS (GETBASE)");
}

q15 = results[spot];
spot++;
trace("q15: " + q15);
if(q15 == 3) {
        q15 = results[spot];
        spot++;
        if(q15 == 3) {
                BaseR = 1;
        }
        else if(q15 == 2) {
                BaseR = .8;
        }
        else {
                BaseR = .6;
        }
}
else if(q15 == 1) {
        q15 = results[spot];
        spot++;
        if(q15 == 3) {
                BaseR = .4;
        }
        else if(q15 == 2) {
                BaseR = .2;
        }
        else {
                BaseR = 0;
        }
}
else if(q15 == 2){
        q15 = results[spot];
        spot++;
        if(q15 == 3) {
                BaseR = .6;
        }
        else if(q15 == 2) {
                BaseR = .5;
        }
        else {
                BaseR = .4;
        }
}
else {
        trace("ERROR IN Q15 ANALYSIS (GETBASE)");
}


q16 = results[spot];
spot++;
trace("q16: " + q16);
if(q16 == 3) {
        q16 = results[spot];
        spot++;
        if(q16 == 3) {
                BaseC = 1;
        }
        else if(q16 == 2) {
                BaseC = .8;
        }
        else {
                BaseC = .6;
        }
```

```
        }
        else if(q16 == 1) {
                q16 = results[spot];
                spot++;
                if(q16 == 3) {
                        BaseC = .4;
                }
                else if(q16 == 2) {
                        BaseC = .2;
                }
                else {
                        BaseC = 0;
                }
        }
        else if(q16 == 2) {
                q16 = results[spot];
                spot++;
                if(q16 == 3) {
                        BaseC = .6;
                }
                else if(q16 == 2) {
                        BaseC = .5;
                }
                else {
                        BaseC = .4;
                }
        }
        else {
                trace("ERROR IN Q16 ANALYSIS (GETBASE)");
        }

        trace("BaseI: " + BaseI);
        trace("BaseR: " + BaseR);
        trace("BaseC: " + BaseC);

}

function getPos() {

        if(BaseI == 0) {
                IBot = 0;
                ITop = .1;
        }
        else if(BaseI == 1) {
                IBot = .9;
                ITop = 1;
        }
        else {
                IBot = BaseI - .1;
                ITop = BaseI + .1;
        }
        if(BaseR == 0) {
                RBot = 0;
                RTop = .1;
        }
        else if(BaseR == 1) {
                RBot = .9;
                RTop = 1;
        }
        else {
                RBot = BaseR - .1;
                RTop = BaseR + .1;
        }
        if(BaseC == 0) {
                CBot = 0;
                CTop = .1;
        }
        else if(BaseC == 1) {
                CBot = .9;
                CTop = 1;
        }
        else {
                CBot = BaseC - .1;
                CTop = BaseC + .1;
        }
}



function getFinal() {
```

```
var avg;
var closer1, closer2;
var diffT, diffB;

if(ILT == 0 && IGT == 0) {
       FinalI = (IBot + ITop)/2;
}
else if(IGT > ITop) {
       avg = (IGT + ITop)/2;
       FinalI = avg;
}
else if(ILT < IBot) {
       avg = (ILT + IBot)/2;
       FinalI = avg;
}
else if(IGT < IBot && ILT > ITop) {
       diffT = ILT - ITop;
       diffB = IBot - IGT;

       if(diffB > diffT) {
              FinalI = IBot;
       }
       else if(diffB == diffT) {
              FinalI = (IBot + ITop)/2;
       }
       else {
              FinalI = ITop;
       }
}

else if( ILT < ITop && IGT < IBot ) {
       FinalI = IBot;
}

else if( IGT < ITop  && ILT > ITop ) {
       FinalI = ITop;
}

else if(IGT > IBot && ILT < ITop) {
        avg = (IBot + ITop)/2;
        closer1 = abs(IGT - avg);
        closer2 = abs(ILT - avg);
        if(closer1 > closer2) {
               FinalI = ILT;
        }
        else if(closer2 > closer1) {
               FinalI = IGT;
        }
        else {
               FinalI = (IGT + ILT)/2;
        }
}


else if(IGT == ITop) {
       FinalI = IGT;
}
else if(ILT == IBot) {
       FinalI = ILT;
}
else if(ILT == ITop && IGT > IBot) {
       avg = (IBot + ITop)/2;
        closer1 = abs(IGT - avg);
        closer2 = abs(ILT - avg);
        if(closer1 > closer2) {
               FinalI = ILT;
        }
        else if(closer2 > closer1) {
               FinalI = IGT;
        }
        else {
               FinalI = (IGT + ILT)/2;
        }
}
else if(ILT == ITop && IGT < IBot) {
       FinalI = IBot;
}
else if(IGT == IBot && ILT > ITop) {
       FinalI = ITop;
}
else if(IGT == IBot && ILT < ITop) {
       avg = (IBot + ITop)/2;
        closer1 = abs(IGT - avg);
        closer2 = abs(ILT - avg);
        if(closer1 > closer2) {
               FinalI = ILT;
        }
        else if(closer2 > closer1) {
               FinalI = IGT;
        }
        else {
               FinalI = (IGT + ILT)/2;
        }
}

else if(IGT == IBot && ILT == ITop) {
       avg = (IBot + ITop)/2;
       FinalI = avg;
}

else {

}


if(RLT == 0 && RGT == 0) {
       FinalR = (RBot + RTop)/2;
}
else if(RGT > RTop) {
       avg = (RGT + RTop)/2;
       FinalR = avg;
}
else if(RLT < RBot) {
       avg = (RLT + RBot)/2;
       FinalR = avg;
}
else if(RGT < RBot && RLT > RTop) {
       diffT = RLT - RTop;
       diffB = RBot - RGT;

       if(diffB > diffT) {
              FinalR = RBot;
       }
       else if(diffB == diffT) {
              FinalR = (RBot + RTop)/2;
       }
       else {
              FinalR = RTop;
       }
}

else if( RLT < RTop && RGT < RBot ) {
       FinalR = RBot;
}

else if( RGT < RTop && RLT > RTop ) {
       FinalR = RTop;
}

else if(RGT > RBot && RLT < RTop) {
        avg = (RBot + RTop)/2;
        closer1 = abs(RGT - avg);
        closer2 = abs(RLT - avg);
        if(closer1 > closer2) {
               FinalR = RLT;
        }
        else if(closer2 > closer1) {
               FinalR = RGT;
        }
        else {
               FinalR = (RGT + RLT)/2;
        }
}


else if(RGT == RTop) {
       FinalR = RGT;
}
else if(RLT == RBot) {
       FinalR = RLT;
}
else if(RLT == RTop && RGT > RBot) {
       avg = (RBot + RTop)/2;
        closer1 = abs(RGT - avg);
```

```
        closer2 = abs(RLT - avg);
        if(closer1 > closer2) {
               FinalR = RLT;
        }
        else if(closer2 > closer1) {
               FinalR = RGT;
        }
        else {
               FinalR = (RGT + RLT)/2;
        }
}
else if(RLT == RTop && RGT < RBot) {
       FinalR = RBot;
}
else if(RGT == RBot && RLT > RTop) {
       FinalR = RTop;
}
else if(RGT == RBot && RLT < RTop) {
       avg = (RBot + RTop)/2;
        closer1 = abs(RGT - avg);
        closer2 = abs(RLT - avg);
        if(closer1 > closer2) {
               FinalR = RLT;
        }
        else if(closer2 > closer1) {
               FinalR = RGT;
        }
        else {
               FinalR = (RGT + RLT)/2;
        }
}
else if(RGT == RBot && RLT == RTop) {
       avg = (RBot + RTop)/2;
       FinalR = avg;
}
else {

}



if(CLT == 0 && CGT == 0) {
       FinalC = (CBot + CTop)/2;
}
else if(CGT > CTop) {
       avg = (CGT + CTop)/2;
       FinalC = avg;
}
else if(CLT < CBot) {
       avg = (CLT + CBot)/2;
       FinalC = avg;
}
else if(CGT < CBot && CLT > CTop) {
       diffT = CLT - CTop;
       diffB = CBot - CGT;

       if(diffB > diffT) {
              FinalC = CBot;
       }
       else if(diffT == diffB) {
              FinalC = (CBot + CTop)/2;
       }
       else {
              FinalC = CTop;
       }
}
else if( CLT < CTop && CGT < CBot ) {
       FinalC = CBot;
}

else if( CGT < CTop && CLT > CTop ) {
       FinalC = CTop;
}

else if(CGT > CBot && CLT < CTop) {
        avg = (CBot + CTop)/2;
        closer1 = abs(CGT - avg);
        closer2 = abs(CLT - avg);
        if(closer1 > closer2) {
               FinalC = CLT;
        }
        else if(closer2 > closer1) {
               FinalC = CGT;
        }
        else {
               FinalC = (CGT + CLT)/2;
        }
}

else if(CGT == CTop) {
       FinalC = CGT;
}
else if(CLT == CBot) {
       FinalC = CLT;
}
else if(CLT == CTop && CGT > CBot) {
       avg = (CBot + CTop)/2;
        closer1 = abs(CGT - avg);
        closer2 = abs(CLT - avg);
        if(closer1 > closer2) {
               FinalC = CLT;
        }
        else if(closer2 > closer1) {
               FinalC = CGT;
        }
        else {
               FinalC = (CGT + CLT)/2;
        }
}
else if(CLT == CTop && CGT < CBot) {
       FinalC = CBot;
}
else if(CGT == CBot && CLT > CTop) {
       FinalC = CTop;
}
else if(CGT == CBot && CLT < CTop) {
       avg = (CBot + CTop)/2;
        closer1 = abs(CGT - avg);
        closer2 = abs(CLT - avg);
        if(closer1 > closer2) {
               FinalC = CLT;
        }
        else if(closer2 > closer1) {
               FinalC = CGT;
        }
        else {
               FinalC = (CGT + CLT)/2;
        }
}
else if(CGT == CBot && CLT == CTop) {
       avg = (CBot + CTop)/2;
       FinalC = avg;
}
else if(CGT == CBot && CLT == CTop) {
       FinalC = CTop -.1;
}
else {

}

}
```

A.3 Walkthrough Results

The walkthrough of the IRCspec tool was used to initially test the feasibility of the scoring algorithm. Figure A.2 depicts results from three different participants (P1, P2, and P3) grading four systems (Info Art was graded twice).

Manual Walkthrough

	System	Est. IRC	MOD1	MOD2a	MOD2b	MOD3	Adj	Base IRC	Possible Range	Final IRC
P1	**Info Art**	.25/ .3/ 1	I< .875 R< .875 C< .875	.125<I<.5	I>.375 .25<R<.75	C>.375	.375<I<.5 .25<R<.75 .375<C<.875	I=.2 R=.2 C=.8	.1<I<.3 .1<R<.3 .7<C<.9	.3/ .25/ .875
	Flowers In bloom	.15/.75/.25	-	.375<I<.75	I>.375 R>.75	-	.375<I<.75 R>.75	I=.2 R=.8 C=.2	.1<I<.3 .7<R<.9 .1<C<.3	.3/ .7/ .2
P2	**Clip-It**	1/1/.5	I>.25 R>.375 C>.375	I>.25 R>.375	I>.25 R>.625	I>.5 C>.75	I>.5 R>.625 C>.75	I=1 R=1 C=.8	I>.9 R>.9 .7<C<.9	1/ 1/ .75
	Bonzi Buddy	.8/.8/.9	I>.5 R>.25 C>.25	.375<I>.75	I>.5 R>.75	C>.375	.5<I<.75 R>.75 C>.375	I=.8 R=.8 C=.6	.7<I<.9 .7<R<.9 .5<C<.7	.75/ .9/ .7
P3	**Info Art II**	.3/ .9/ .5	I<.75	.375<I<.75	I>.375 .25<R<.75	I<.375 C<.75	I=.375 .25<R<.75 C<.75	I=.4 R=.6 C=.2	.3<I<.5 .5<R<.7 .1<C<.3	.375/ .5/ .3

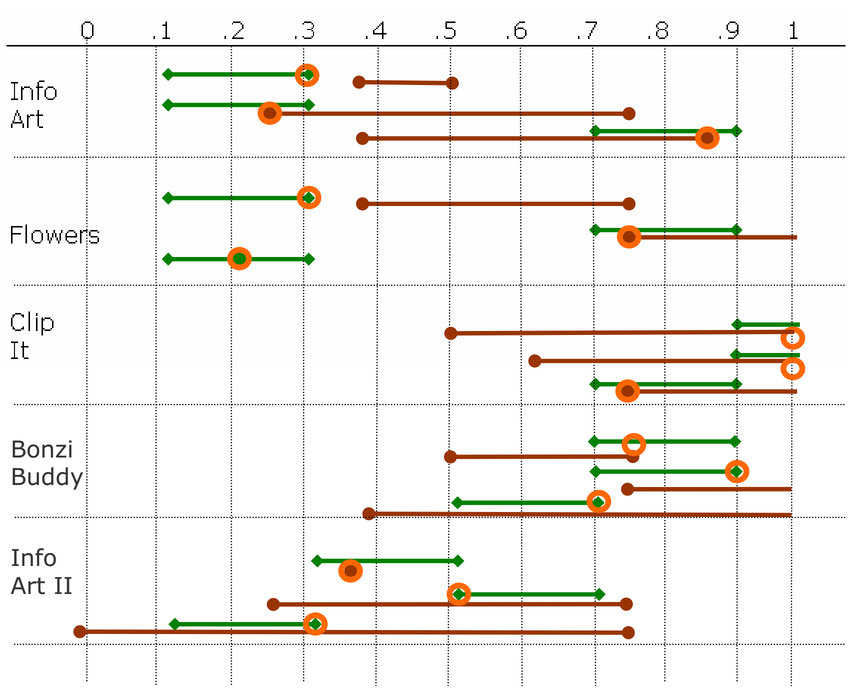


Figure A.2: Results of the manual walkthrough.

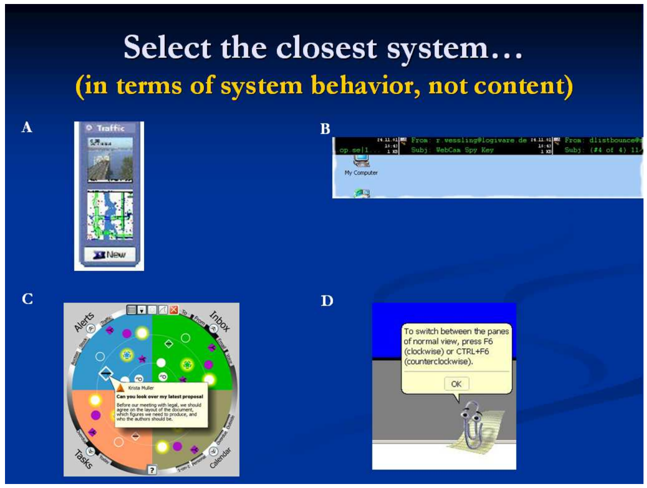


Figure A.3: System choices for Problem 1.

A.4 Experiment Materials

A.4.1 Design Problems

Problem 1. *You have been asked to design a desktop notification system that provides* ***sport score updates*** *for several games that users select. You anticipate that users (probably typical college students) will want to glance at this system quite frequently during a course of several hours, as they perform other desktop processing tasks. These primary tasks include word processing, making presentations, chat, and surfing the Internet. Although you feel it will be important for the notification system to always be visible, you don't think it should take up much screen space or be overly distracting. You don't think that users will usually want to click on anything to receive updatesbut it is possible they theyll want to use the system to launch to more details about close scores or important games. However, you guess that most users will just want to know the scores.*

Description of System Choices—Problem 1.

- **System A.** A small window of a continuously updated live video or pictorial representation of information status that would reside within a desktop sidebar.
- **System B.** A ticker-like interface that scrolls information across a set part of the screen.
- **System C.** A small interface that sits in the corner that is organized into categories of information and allows pulsing or flashing icon symbols to appear when new information items are available.
- **System D.** An agent that pops up to deliver messages that may be of interest.

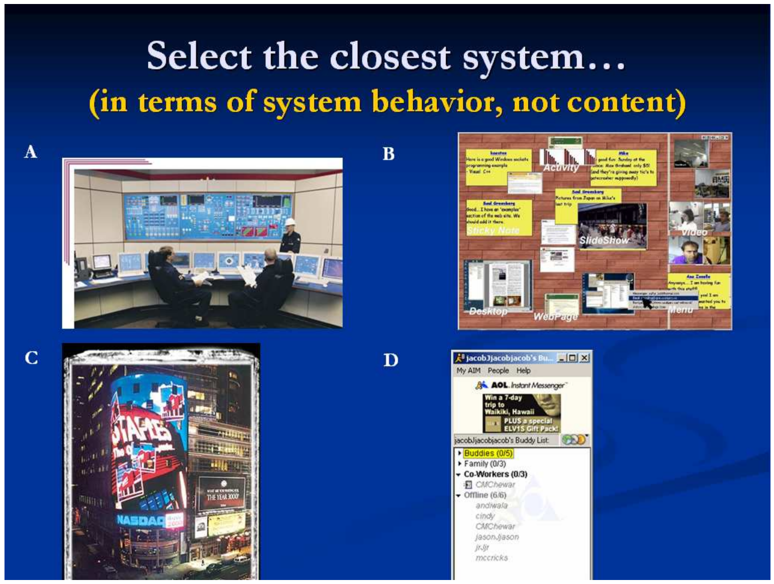


Figure A.4: System choices for Problem 2.

Problem 2. *You want to design a notification system that can be used in a typical classroom, perhaps mounted on a wall. The system will provide information to teachers about the **progress of students** that are working on a project in groups. The system should show key deadlines to remind teachers and students about the urgency of their current tasks. As they submit electronic work products and chat with team members in other classrooms, these activities should be depicted (real time) on the system. Teachers should be able to glance at this system while they are busy with work like grading, making slides, and creating lesson plans. Their primary concern is facilitating the groupwork, although they do not want to interfere needlessly. Teachers should be able to compare the progress of different groups at any instant to distinguish the groups that need help. This system should also allow them to gain an overall understanding of the progress of the class.*

Description of System Choices—Problem 2.

- **System A.** A display that is organized into various charts that must be carefully studied to understand information status, but conveys patterns and historical data trends.
- **System B.** A display that resembles a bulletin board, allows collection of notes, and chimes whenever a new message is posted.
- **System C.** A display like an advertising billboard that cycles through hi-fidelity announcements and status summaries.
- **System D.** An enlarged version of a simple status indicator, a display that tells you if your buddies are logged on or not.

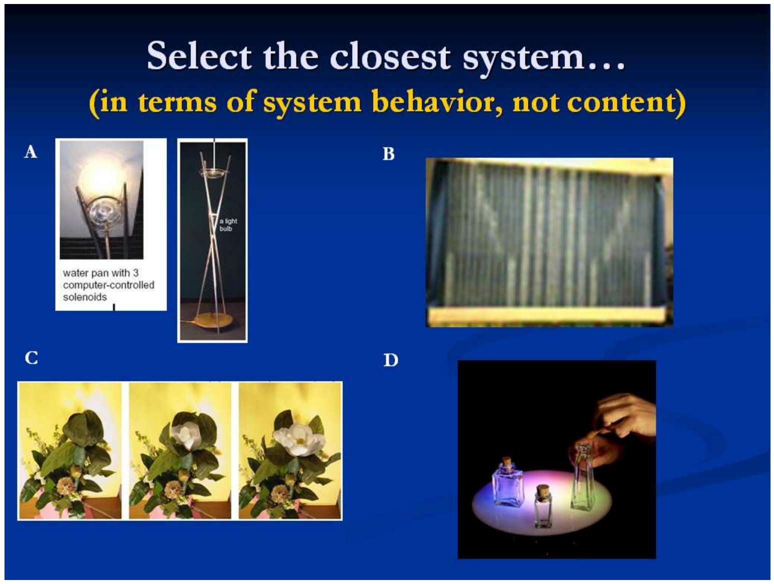


Figure A.5: System choices for Problem 3.

Problem 3. *The notification system that you are trying to design is for working mothers in a particular organization that have children in that organizations daycare. While the mothers perform a variety of desktop processing tasks, especially data entry and customer service phone calls, your system would provide almost subliminal awareness of the* ***childs current activities****. These activities (such as sleeping, eating, playing, drawing) are updated through a network by daycare staff and will be depicted on with your system. However, the organization (who is paying for these systems) demands an interface that is very subtle, entirely non distracting, and unnoticeable to clients or other employees that would visit the mothers office space.*

Description of System Choices—Problem 3.

- **System A.** An elegantly designed apparatus that projects light through water that is rippled according to information changes.
- **System B.** A large collection of tubes that allow bubbles to form meaningful, yet abstract patterns according to some information source.
- **System C.** An artificial flower or arrangement of flowers that slowly open and close according an information source.
- **System D.** Pleasingly designed bottles that, when left open, emit natural, background sounds (such as the sound of rain or waves on a beach)—with the intensity or volume corresponding to an information source.

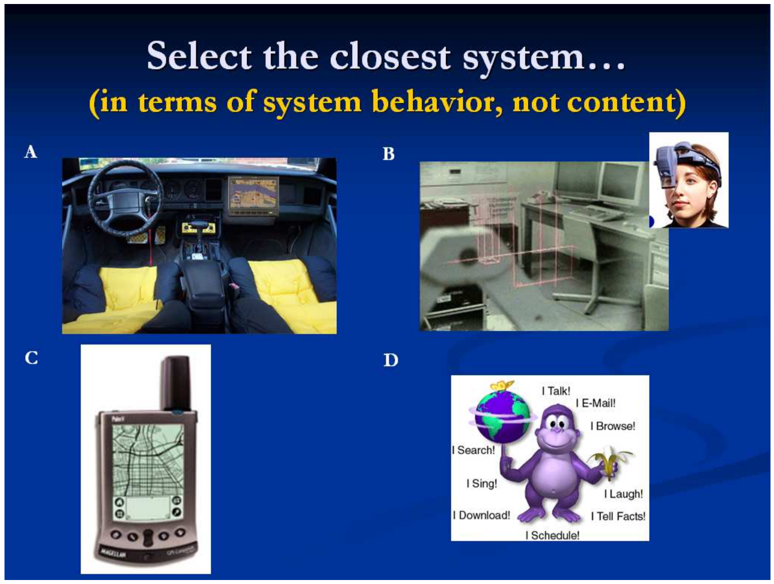


Figure A.6: System choices for Problem 4.

Problem 4. *Your development team has been contracted to design a wearable display for emergency personnel, such as policemen and firemen, that would be* ***used during emergency operations*** *(i.e., fires and rescue missions). Emergency personnel will use this notification system as they receive instructions from supervisors and quickly find their way around an often dangerous scene of emergency, which could be a complicated school or office building. The system you are designing (which they will attach to protective eyeglasses) will display maps and urgent textual information related to specific areas that they physically move through. Since the emergency personnel will have a GPS (global positioning system) sensor, the system will be able to provide essential information for the immediate physical surroundings.*

Description of System Choices—Problem 4.

- **System A.** A system like an in-vehicle navigation system that is programmed and studied at the start of a trip, very briefly glanced at while the vehicle is moving, but beeps at you to turn at certain points.
- **System B.** A system that superimposes subtle, simple lines or text over the user's field of vision to convey a single type of information.
- **System C.** A handheld display that depicts a full version of a map that updates according to the user's location.
- **System D.** A pop-up type system (implemented as a head-mounted or handheld display) that gets the full attention of the user to immediately deliver information.

A.4.2 Manual IRC Calculation Instructions

Figure A.7 shows the instructions given to experiment participants that provided design model IRC ratings without using the IRCspec tool. These instructions represented the working definitions of interruption, reaction, and comprehension.

A) **How much of a desire** will your users have for their attention to be reallocated from their primary task to the notification by the system?

Things to consider in your rating:

- How often notifications will be delivered & valued
- The impact of not noticing a notification
- The impact of distracting the primary task
- Relative importance between the primary task and notification

Rate this as any number between zero and one, where:
0 = **none** (or very low), **.5** = moderate, and **1 = very high**

B) How important will reacting to the notification be?

Things to consider in your rating:

- What is the likely or possible reaction to the notification information
- How important is the reaction relative to the continuation of the primary task
- What is the consequence of not reacting properly or quicky
- How frequently will users receive a notification and decide to do nothing

Rate this as any number between zero and one, where:
0 = none (or very low), **.5** = moderate, and **1 = very high**

C) How much will users value making sense of and then remembering the notification content?

Things to consider in your rating:

- Will users want to detect trends or patterns in the information
- After monitoring the system for several hours, will users want to understand the information better
- Is prior knowledge important to understand the meaning of the notifications

Rate this as any number between zero and one, where:
0 = none (or very low), **.5** = moderate, and **1 = very high**

Figure A.7: Instructions provided to participants who did not use the IRCspec tool.

A.4.3 Response Forms

Participant ID______

Please indicate your rating answers here:

<u>Design Problem 1</u>

System selection _________ (A, B, C, or D)

<u>Design Problem 2</u>

System selection _________ (A, B, C, or D)

<u>Design Problem 3</u>

System selection _________ (A, B, C, or D)

<u>Design Problem 4</u>

System selection _________ (A, B, C, or D)

Figure A.8: Response form used by participants using the IRCspec tool. On this form, the participants indicated their choice for the system that would the best starting point for thinking about the design problem.

Participant ID______

Please indicate your rating answers here:

Design Problem 1

Rating: A = (0 – 1)
B = (0 – 1)
C = (0 – 1)

System selection _________ (A, B, C, or D)

Design Problem 2

Rating: A = (0 – 1)
B = (0 – 1)
C = (0 – 1)

System selection _________ (A, B, C, or D)

Design Problem 3

Rating: A = (0 – 1)
B = (0 – 1)
C = (0 – 1)

System selection _________ (A, B, C, or D)

Design Problem 4

Rating: A = (0 – 1)
B = (0 – 1)
C = (0 – 1)

System selection _________ (A, B, C, or D)

Figure A.9: Response form used by participants (not using the IRCspec tool) to record IRC rating assessments.

A.5 Experiment Data and Analysis

	Expert 1	Expert 2	*Diff.*
Design Problem 1			
Manual Method	.4/.4/.7	.2/.1/.8	*.2/.3/.1*
IRCspec Method	.5/.5/.7	.4/.5/.7	*.1/ 0/ 0*
Diff.	*.1/.1/0*	*.2/**.4**/.1*	
Design Problem 2			
Manual Method	.6/.6/.8	.8/.8/.5	*.2/.2/.3*
IRCspec Method	.6/.6/.7	.6/.6/.8	*0/ 0/.1*
Diff.	*0/ 0/.1*	*.2/.2/.3*	
Design Problem 3			
Manual Method	.1/.1/.3	0/.1/.5	*.1/ 0/.2*
IRCspec Method	.3/.1/.5	.2/.4/.6	*.1/.3/.1*
Diff.	*.2/ 0/.2*	*.2/.3/.1*	
Design Problem 4			
Manual Method	.6/.8/.4	.9/.9/1	*.3/.1/.3*
IRCspec Method	.7/.7/.5	.5/.8/.7	*.2/.1/.2*
Diff.	*.1/.1/.1*	***.4**/.1/.3*	
Avg Diffs.	*.10/.05/.10*	*.25/.25/.20*	

Manual Method	=	*.20/.15/.23*
IRCspec Method	=	***.10/.10/.10***

Figure A.10: Comparison of IRC-rating consistency between two experts. Both provided manual IRC estimations before using the IRCspec tool. Neither expert had used the tool before the experiment. Parameters calculated using IRCspec were not seen by either expert during the experiment.

Kendall's correlation - Expert ratings

Parameter	Design Prob.	Manual Expert 1	Manual Expert 2	IRCspec Expert 1	IRCspec Expert 2
I	1	0.4	0.2	0.5	0.4
	2	0.6	0.8	0.6	0.6
	3	0.1	0	0.3	0.2
	4	0.6	0.9	0.7	0.5
R	1	0.4	0.1	0.5	0.5
	2	0.6	0.8	0.6	0.6
	3	0.1	0.1	0.1	0.4
	4	0.8	0.9	0.7	0.8
C	1	0.7	0.8	0.7	0.7
	2	0.8	0.5	0.7	0.8
	3	0.3	0.5	0.5	0.6
	4	0.4	1	0.5	0.7
Tau:		0.45		**0.59**	
P-value:		0.0546		**0.013**	

Figure A.11: Interrater reliability of IRC-rating consistency between two experts, calculated using Kendall's tau.

Manual	Problem 1			Problem 2			Problem 3			Problem 4			
ID	I	R	C	I	R	C	I	R	C	I	R	C	
1	0.7	0.5	0.5	0.8	0.7	0.5	0.1	0.1	0.2	1	1	0.7	
2	0.25	0.5	0.5	0.2	0.1	0.8	0.5	1	1	1	1	1	
3	0.9	0.2	0.8	1	1	0.5	0.3	0.5	0.2	1	1	0.7	
4	0.8	0.1	0.4	0.9	0.8	0.9	0.4	1	0.2	0.6	0.75	0.9	
5	0.2	0	0.6	0.7	0.4	0.9	0.1	0.1	0.2	0.5	1	1	
6	1	0.3	0.1	0.8	0.7	0.5	0.7	0.2	0	0.5	1	0.6	
7	0.3	0.2	0	0.5	0.1	0.8	0.5	0.2	0	0.8	0.7	0.2	
8	0.7	0.1	0.5	0	0.7	1	0	0.2	0.2	0.4	1	0.2	
9	0.5	0.2	0.6	0.3	0.1	0.9	0.2	0.5	0.1	0.9	1	0.8	
10	0.3	0.2	0.6	0.5	0.5	0.8	0.2	0.5	0.7	0.8	0.9	0.8	
avg	0.57	0.23	0.46	0.57	0.51	0.76	0.3	0.43	0.28	0.75	0.94	0.69	
stdev	**0.29**	0.16	0.24	**0.33**	**0.32**	0.19	0.22	**0.34**	**0.32**	0.23	0.12	**0.29**	0.255
IRCspec													
21	0.5	0.7	0.5	0.6	0.5	0.5	0.83	0.83	0.9	0.83	0.83	0.9	
22	0.83	0.8	0.5				0.5	0.63	0.5	0.63	0.83	0.6	
23	0.68	0.7	0.5	0.7	0.5	0.9	0.5	0.1	0.4	0.68	0.5	0.7	
24	0.7	0.58	0.9	0.5	0.7	0.7	0.46	0.28	0.6	0.7	0.83	0.61	
25	0.61	0.5	0.3	0.68	0.5	0.6	0.61	0.5	0.6	0.81	0.83	0.6	
26	0.54	0.5	0.5	0.6	0.7	0.5	0.6	0.28	0.6	0.5	0.9	0.7	
27	0.4	0.5	0.3	0.62	0.5	0.6	0.34	0.1	0.4	0.61	0.7	0.6	
28	0.5	0.5	0.5	0.6	0.5	0.5	0.6	0.5	0.5	0.5	0.73	0.7	
29	0.53	0.18	0.4	0.61	0.68	0.6	0.34	0.33	0.6	0.9	0.9	0.9	
30	0.6	0.5	0.7				0.81	0.7	0.6	0.61	0.83	0.7	
avg	0.59	0.55	0.51	0.61	0.57	0.61	0.56	0.42	0.57	0.68	0.79	0.7	
stdev	0.12	0.17	0.18	0.06	0.1	0.14	0.17	0.25	0.14	0.14	0.12	0.11	0.141
Expert 1													
manual	0.4	0.4	0.7	0.6	0.6	0.8	0.1	0.1	0.3	0.6	0.8	0.4	
IRCspec	0.5	0.5	0.7	0.6	0.6	0.7	0.3	0.1	0.5	0.7	0.7	0.5	

+/- .15 agreement = [green] = greater than +/- .3 away [pink]

	+/- .15 agreement: *Total*			greater than +/- .3 away: *Total*	
Manual Method:	45	38%	#	31	26%
IRCspec:	65	59%	#	15	14%

Figure A.12: Raw IRC-ratings provided by the participants using each of the two methods (manual and IRCspec). Cell shading shows the general difference between ratings and the expert score (calculated with the same method).

ID	I	R	C	I	R	C	I	R	C	I	R	C	
Manual Diff. from Expert 1													
1	0.30	0.10	0.20	0.20	0.10	0.30	0.00	0.00	0.10	**0.40**	0.20	0.30	
2	0.15	0.10	0.20	**0.40**	**0.50**	0.00	**0.40**	**0.90**	**0.70**	**0.40**	0.20	**0.60**	
3	**0.50**	0.20	0.10	**0.40**	**0.40**	0.30	0.20	**0.40**	0.10	**0.40**	0.20	0.30	
4	**0.40**	0.30	0.30	0.30	0.20	0.10	0.30	**0.90**	0.10	0.00	0.05	**0.50**	
5	0.20	**0.40**	0.10	0.10	0.20	0.10	0.00	0.00	0.10	0.10	0.20	**0.60**	
6	**0.60**	0.10	**0.60**	0.20	0.10	0.30	**0.60**	0.10	0.30	0.10	0.20	0.20	
7	0.10	0.20	**0.70**	0.10	**0.50**	0.00	**0.40**	0.10	0.30	0.20	0.10	0.20	
8	0.30	0.30	0.20	**0.60**	0.10	0.20	0.10	0.10	0.10	0.20	0.20	0.20	
9	0.10	0.20	0.10	0.30	**0.50**	0.10	0.10	**0.40**	0.20	0.30	0.20	**0.40**	
10	0.10	0.20	0.10	0.10	0.10	0.00	0.10	**0.40**	**0.40**	0.20	0.10	**0.40**	
avg	0.28	0.21	0.26	0.27	0.27	0.14	0.22	0.33	0.24	0.23	0.17	0.37	0.25
IRCspec Diff. from Expert 1													
21	0.00	0.20	0.20	0.00	0.10	0.20	**0.53**	**0.73**	**0.40**	0.13	0.13	**0.40**	
22	**0.33**	0.30	0.20				0.20	**0.53**	0.00	0.08	0.13	0.10	
23	0.18	0.20	0.20	0.10	0.10	0.20	0.20	0.00	0.10	0.02	0.20	0.20	
24	0.20	0.08	0.20	0.10	0.10	0.00	0.16	0.18	0.10	0.00	0.13	0.11	
25	0.11	0.00	**0.40**	0.08	0.10	0.10	**0.31**	**0.40**	0.10	0.11	0.13	0.10	
26	0.04	0.00	0.20	0.00	0.10	0.20	0.30	0.18	0.10	0.20	0.20	0.20	
27	0.10	0.00	**0.40**	0.02	0.10	0.10	0.04	0.00	0.10	0.09	0.00	0.10	
28	0.00	0.00	0.20	0.00	0.10	0.20	0.30	**0.40**	0.00	0.20	0.03	0.20	
29	0.03	**0.33**	0.30	0.01	0.08	0.10	0.04	0.23	0.10	0.20	0.20	**0.40**	
30	0.10	0.00	0.00				**0.51**	**0.60**	0.10	0.09	0.13	0.20	
avg	0.11	0.11	0.23	0.04	0.10	0.14	0.26	0.32	0.11	0.11	0.13	0.20	0.16

Figure A.13: Absolute differences between participants IRC ratings and expert ratings (calculated with the same method). Differences greater than 0.3 are highlighted.

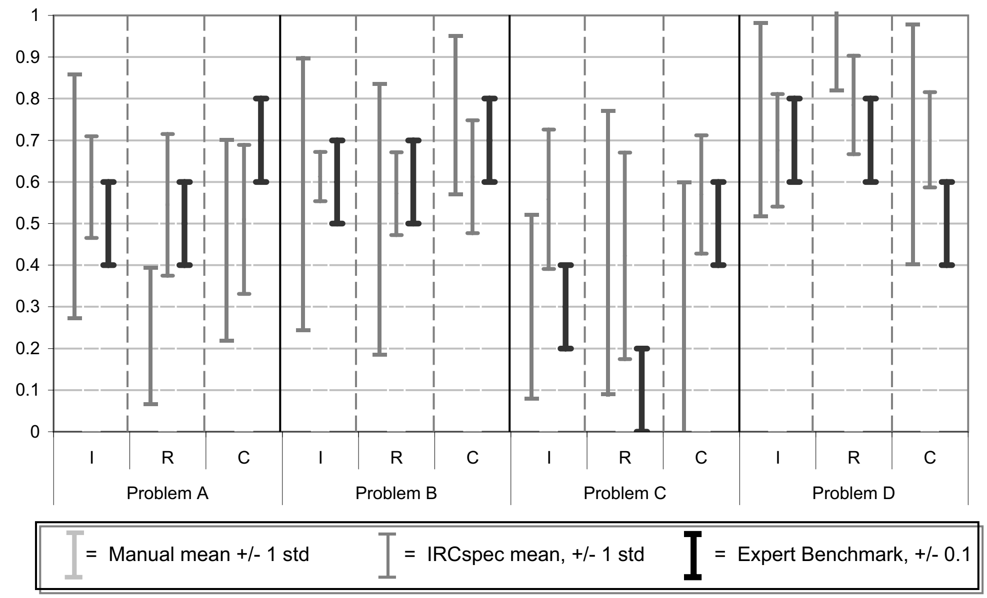


Figure A.14: Rating ranges for each of the four design problems. Expert Benchmark scores are shown with a +/- 0.1 range, and participant ratings means are shown with +/- one standard deviation. From this depiction, it can be inferred that most participants using the IRCspec tool would hit within the expert range, except in four cases (75%): C-value for Problem A, I-value and R-value for Problem C, and the C-value for Problem D. On the contrary, most participants using the Manual Method would only hit with the expert range in two cases (12.5%): C-value for Problem B and I-value for Problem C.

Cronbach's Coefficient Alpha

Raw Scores

Manual Method

	I	R	C	Total	Cronbach's
Prob 1	0.7	0.5	0.5	1.7	
	0.25	0.5	0.5	1.25	
	0.9	0.2	0.8	1.9	
	0.8	0.1	0.4	1.3	
	0.2	0	0.6	0.8	
	1	0.3	0.1	1.4	
	0.3	0.2	0	0.5	
	0.7	0.1	0.5	1.3	
	0.5	0.2	0.6	1.3	
	0.3	0.2	0.6	1.1	
var:	0.09	0.03	0.06	0.16	-0.10
Prob 2	0.8	0.7	0.5	2	
	0.2	0.1	0.8	1.1	
	1	1	0.5	2.5	
	0.9	0.8	0.9	2.6	
	0.7	0.4	0.9	2	
	0.8	0.7	0.5	2	
	0.5	0.1	0.8	1.4	
	0	0.7	1	1.7	
	0.3	0.1	0.9	1.3	
	0.5	0.5	0.8	1.8	
var:	0.11	0.11	0.04	0.24	-0.06
Prob 3	0.1	0.1	0.2	0.4	
	0.5	1	1	2.5	
	0.3	0.5	0.2	1	
	0.4	1	0.2	1.6	
	0.1	0.1	0.2	0.4	
	0.7	0.2	0	0.9	
	0.5	0.2	0	0.7	
	0	0.2	0.2	0.4	
	0.2	0.5	0.1	0.8	
	0.2	0.5	0.7	1.4	
var:	0.05	0.12	0.10	0.44	0.60
Prob 4	1	1	0.7	2.7	
	1	1	1	3	
	1	1	0.7	2.7	
	0.6	0.75	0.9	2.25	
	0.5	1	1	2.5	
	0.5	1	0.6	2.1	
	0.8	0.7	0.2	1.7	
	0.4	1	0.2	1.6	
	0.9	1	0.8	2.7	
	0.8	0.9	0.8	2.5	
var:	0.05	0.01	0.08	0.21	0.42
				Average:	0.22

IRCspec Method

	I	R	C	Total	Cronbach's
Prob 1	0.5	0.7	0.5	1.7	
	0.825	0.8	0.5	2.125	
	0.675	0.7	0.5	1.875	
	0.7	0.575	0.9	2.175	
	0.6125	0.5	0.3	1.4125	
	0.5375	0.5	0.5	1.5375	
	0.4	0.5	0.3	1.2	
	0.5	0.5	0.5	1.5	
	0.525	0.175	0.4	1.1	
	0.6	0.5	0.7	1.8	
	0.01	0.03	0.03	0.13	0.62
Prob 2	0.6	0.5	0.5	1.6	
	0.7	0.5	0.9	2.1	
	0.5	0.7	0.7	1.9	
	0.675	0.5	0.6	1.775	
	0.6	0.7	0.5	1.8	
	0.615	0.5	0.6	1.715	
	0.6	0.5	0.5	1.6	
	0.6125	0.675	0.6	1.8875	
	0.00	0.01	0.02	0.03	-0.21
Prob 3	0.825	0.825	0.9	2.55	
	0.5	0.625	0.5	1.625	
	0.5	0.1	0.4	1	
	0.4625	0.275	0.6	1.3375	
	0.6125	0.5	0.6	1.7125	
	0.6	0.275	0.6	1.475	
	0.3375	0.1	0.4	0.8375	
	0.6	0.5	0.5	1.6	
	0.3375	0.325	0.6	1.2625	
	0.806	0.7	0.6	2.106	
	0.03	0.06	0.02	0.25	0.85
Prob 4	0.825	0.825	0.9	2.55	
	0.625	0.825	0.6	2.05	
	0.675	0.5	0.7	1.875	
	0.7	0.825	0.6125	2.1375	
	0.8125	0.825	0.6	2.2375	
	0.5	0.9	0.7	2.1	
	0.6115	0.7	0.6	1.9115	
	0.5	0.725	0.7	1.925	
	0.9	0.9	0.9	2.7	
	0.6115	0.825	0.7	2.1365	
	0.02	0.01	0.01	0.07	0.58
					0.69

Figure A.15: Interrater reliability of raw data.

Cronbach's Coefficient Alpha

Difference from Expert 1

Manual Method

I	R	C	Total	Cronbach's
0.30	0.10	-0.20	0.20	
-0.15	0.10	-0.20	-0.25	
0.50	-0.20	0.10	0.40	
0.40	-0.30	-0.30	-0.20	
-0.20	-0.40	-0.10	-0.70	
0.60	-0.10	-0.60	-0.10	
-0.10	-0.20	-0.70	-1.00	
0.30	-0.30	-0.20	-0.20	
0.10	-0.20	-0.10	-0.20	
-0.10	-0.20	-0.10	-0.40	
0.20	0.10	-0.30	0.00	
-0.40	-0.50	0.00	-0.90	
0.40	0.40	-0.30	0.50	
0.30	0.20	0.10	0.60	
0.10	-0.20	0.10	0.00	
0.20	0.10	-0.30	0.00	
-0.10	-0.50	0.00	-0.60	
-0.60	0.10	0.20	-0.30	
-0.30	-0.50	0.10	-0.70	
-0.10	-0.10	0.00	-0.20	
0.00	0.00	-0.10	-0.10	
0.40	0.90	0.70	2.00	
0.20	0.40	-0.10	0.50	
0.30	0.90	-0.10	1.10	
0.00	0.00	-0.10	-0.10	
0.60	0.10	-0.30	0.40	
0.40	0.10	-0.30	0.20	
-0.10	0.10	-0.10	-0.10	
0.10	0.40	-0.20	0.30	
0.10	0.40	0.40	0.90	
0.40	0.20	0.30	0.90	
0.40	0.20	0.60	1.20	
0.40	0.20	0.30	0.90	
0.00	-0.05	0.50	0.45	
-0.10	0.20	0.60	0.70	
-0.10	0.20	0.20	0.30	
0.20	-0.10	-0.20	-0.10	
-0.20	0.20	-0.20	-0.20	
0.30	0.20	0.40	0.90	
0.20	0.10	0.40	0.70	
0.08	0.10	0.10	0.39	0.42

IRCspec Method

I	R	C	Total	Cronbach's
0.00	0.20	-0.20	0.00	
0.33	0.30	-0.20	0.43	
0.18	0.20	-0.20	0.18	
0.20	0.08	0.20	0.48	
0.11	0.00	-0.40	-0.29	
0.04	0.00	-0.20	-0.16	
-0.10	0.00	-0.40	-0.50	
0.00	0.00	-0.20	-0.20	
0.03	-0.33	-0.30	-0.60	
0.10	0.00	0.00	0.10	
0.00	-0.10	-0.20	-0.30	
0.10	-0.10	0.20	0.20	
-0.10	0.10	0.00	0.00	
0.08	-0.10	-0.10	-0.13	
0.00	0.10	-0.20	-0.10	
0.02	-0.10	-0.10	-0.19	
0.00	-0.10	-0.20	-0.30	
0.01	0.08	-0.10	-0.01	
0.53	0.73	0.40	1.65	
0.20	0.53	0.00	0.73	
0.20	0.00	-0.10	0.10	
0.16	0.18	0.10	0.44	
0.31	0.40	0.10	0.81	
0.30	0.18	0.10	0.58	
0.04	0.00	-0.10	-0.06	
0.30	0.40	0.00	0.70	
0.04	0.23	0.10	0.36	
0.51	0.60	0.10	1.21	
0.13	0.13	0.40	0.65	
-0.08	0.13	0.10	0.15	
-0.02	-0.20	0.20	-0.02	
0.00	0.13	0.11	0.24	
0.11	0.13	0.10	0.34	
-0.20	0.20	0.20	0.20	
-0.09	0.00	0.10	0.01	
-0.20	0.03	0.20	0.03	
0.20	0.20	0.40	0.80	
-0.09	0.13	0.20	0.24	
0.03	0.04	0.04	0.21	0.67

Figure A.16: Interrater reliability of difference scores between participants and Expert 1.

Appendix B

IRCresults Documentation, Testing Materials, and Results

B.1 Equation Variables and Behavior

Table B.1: *Relationship between questions used in the IRCspec tool and IRC equation variables, showing values used for answer scores. Table continues on the next page.*

IRCspec Question	**Choices**	**IRC concrete term**	**Effect**
(4) On average, the user's primary task is important.	Strongly agree Agree Neutral Disagree Strongly disagree	(Cost)	co = 1 co = .75 co = .5 co = .25 co = 0
(5) For the user, an accurate interpretation of the notification is important.	Strongly agree Agree Neutral Disagree Strongly disagree	Hit rate	h = 1 h = .75 h = .5 h = .25 h = 0
(6) Which statement describes the general relationship between the importance of the primary task and receiving the notification?	Ptask always more important Ptask important more often Equal importance Notification important more often Notification always more important	Cost of interruption	COI = 1 COI = .75 COI = .5 COI = .25 COI = 0
(7) How can the time interval be characterized?	Regular Sporadic	Response time; hit rate	
(7a) How often will the user expect [regular] notification deliveries?	Continuous (10 sec or less) Every few seconds Every few minutes About once an hour After a few hours About once a day	Response time	t = 1 t = .8 t = .6 t = .4 t = .2 t = 0
(7b) What prompts the [sporadic] notification delivery?	A primary task event or state An external secondary event A response to a user's request	Response time	t = .75 t = .25 t = .5
(8) What percentage of the notification deliveries will be valued by the users?	85% or more 65-85% 35-65% 15-35% less than 15%	Expected perception rate	p = 1 p = .75 p = .5 p = .25 p = 0
(9) Which of the following outcomes are possible for a user after receiving and interpreting a notification while engaged in a primary task?	(1) continue the primary task unchanged (2) perform the primary task differently (3) switch focus to a different primary task 1 and 2 1 and 3 2 and 3 all	Base comp; Hr & Pr; Rt, Sust	c + .25 h + .25, p + .25 t + .25, s - .25 c + .25, h + .25, p + .25 c + .25, t + .25 h + .25, p + .25, t + .25 c + .25, h + .25, p + .25, t + .25

(10) Aside from the value of the information delivered, to what extent will the notification satisfy and please the user?	Very much Somewhat Very little None	[satisfaction]	
(11) What portion of other notifications presented will users want to remember in order to understand patterns and trends?	5 - 80%+ 4 - 50-80% 3 - 20-50% 2 - less than 20%	Projection	f = 1 f = .66 f = .33 f = 0
(12) Based on the complexity of the notification, what is the minimum amount of attention required by the user to understand the meaning?	4 - full attention 3 - evenly divided 2 - little attention 1 - no attention	Primary task sustainment	s = 0 s = .5 s = .75 s = 1
(13) Out of ten average notification deliveries, how many times will users need to relate notification content to existing or prior knowledge in order to understand it?	5 - 8-10 times 4 - 5-8 times 3 - 2-5 times 2 - less than twice	Base comprehension	c = 1 c = .66 c = .33 c = 0

Table B.2: *Concrete terms used in the I, R, and C equations, and usability evaluation assessment techniques for each.*

Concrete Term		Assessment Technique	
Symbol	**Description**	**Analytical/Subjective**	**Empirical/Objective**
COI	cost of interruption	*Given the nature and importance of the user's primary task at the receipt of the notification, how* ***costly*** *would an interruption be?* {extremely = 1; very = .75; moderately = .5; not very = .25; not at all = 0}	Interruption Workbench output; P(High) is weighed at 1, P(Med) = .5, P(Low)=0
s	primary task sustainment	*Compared to the primary task performance before the notification delivery, how much does the primary task performance* ***reduce*** *when the notification is present?* {not at all = 1; less than half = .75; about half = .5; more than half = .25; completely stops = 0}	Ptask performance while multitasking divided by ptask performance as a solo-task
h	hit rate	*How often will users actually notice important changes in the notification, as opposed to not noticing them?* {always = 1; more than half = .75; about half = .5; less than half = .25; never = .0001}	As in Signal Detection Theory, P(H) divided by total signals
t	response time	*In cases where a notification suggests an action for a user to take, how does the user's response time compare to the reasonably desired response time?* {better or as good as expected = 1; slightly slower = .75; about twice as slow as expected = .5; much slower = .25; extremely slow or action never taken; no action ever required = 0}	Determine *actual response time* (*a*) as the difference between signal presentation and signal response; *expected response time* (*e*) provided in system specification; $t = e / a$, when $a > e$ (otherwise $t = 1$)
p	perception rate	*When considering the total number of times a user interacts with the notification system, what is the* ***ratio*** *of the interactions in response to an important notification vs. total interactions (including those when no actual notification was being delivered, i.e., user checking on their own or thinking there was a notification)?* {1 to 1 = 1; 2 to 3= .75; 1 to 2 = .5; 1 to 4 = .25; more than 1 to 4 = 0}	As in Signal Detection Theory, P(H) divided by total responses
c	base comprehension	*How much of the notification content will the user want to remember* <u>*and*</u> *be able to remember several minutes after the notification is delivered?* {all content = 1; more than half = .75; about half = .5; less than half = .25; none at all = 0}	Quiz user on a sample of notification content questions to assess correct interpretation, relationship to goals, and storage in long term memory. Use % correct.
f	projection	*Based on the notification content, how successful will the user be in making projections or predictions about future trends or the long-term state of the system being monitored?* {extremely successful = 1; very successful = .75; somewhat successful = .5; not very successful = .25; not a goal for this system = 0}	Quiz user based on a sample of interpretations that can be projected to predict future states or notification patterns. Use % correct.

Interruption: $\mathbf{I} = 1 - s^{3 \cdot COI}$

Justification

- *I* describes both the appropriateness of an interruption, as well as the actual interruptive effect of the notification artifact (distraction to the primary task). Therefore, "low I" can describe either an artifact that supports attention grading/parallel processing during the performance of an urgent primary task (high sustainment, regardless of COI) or any quality of multitasking performance in a non urgent situation (low COI, regardless of sustainment).

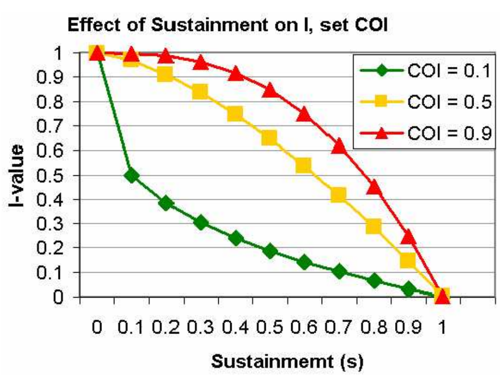
- *Appropriateness of an interruption* is represented by COI (cost of interruption), characterizing the user's willingness to accept an interruption, and thus the urgency of the primary task can be inferred. As established by Horvitz's Interruption Workbench [4], COI describes a total task situation in terms of how much a given user would typically pay in dollars not to be interrupted. The Interruption Workbench records a variety of situation characteristics, such as the specific primary task application, level of ambient noise, recent keystroke and mouse activity, etc) over an extended period of normal user activity. The tool segments the observations into periods in which the task variable combinations are consistent. Users rate each segment, assigning the dollar value they would pay to avoid interruption, allowing Bayesian inference networks to aggregate samples and determine probability distributions for various costs of interruption levels.

- *Actual interruptive effect* can be gauged by primary task sustainment—a metric used to quantify the change in the primary task performance from solo-task performance to dual-task performance.

- The equation is modeled with an exponential COI to reinforce the importance of this factor, but tripled to ensure a fairly wide range of I-values for a given COI and to produce a moderately high I-value (0.65) when both s and COI = 0.5.

Sample results:

COI	s	I	Rationale
0.1	0.9	0.03	Very low cost, very low actual interruption—likely to employ **discrete attention division**
0.1	0.5	0.19	
0.1	0.1	0.50	Classic case of **user initiated interruption**
0.3	0.9	0.09	
0.3	0.5	0.46	
0.3	0.1	0.87	User initiated interruption
0.5	0.9	0.15	Moderately important primary task effectively preserved, combination of graded and discrete attention division
0.5	0.5	0.65	
0.5	0.1	0.97	User initiated interruption possible
0.7	0.9	0.20	
0.7	0.5	0.77	
0.7	0.1	0.99	User initiated interruption possible, but not likely
0.9	0.9	0.25	Urgent primary task effectively preserved, probably through **graded attention division**
0.9	0.5	0.85	
0.9	0.1	0.99	Urgent primary task almost completely interrupted—most likely an extremely valued notification or a design flaw



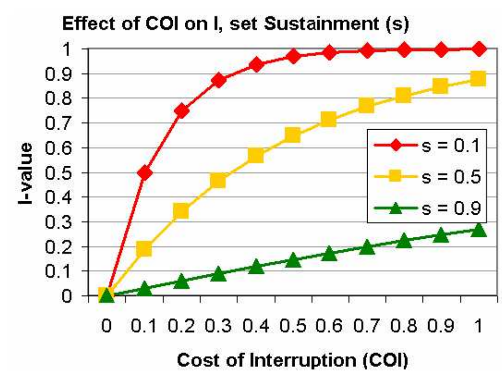


Figure B.1: Summarized justification and sample results for the I equation.

Reaction: $R = \frac{(t \cdot h)^{\frac{1}{3 \cdot COI}}}{2} + \frac{h(0.5 + COI)}{3}$

Justification

- The R equation consists of two parts, each worth up to an R-value component of 0.5. The first term takes two reaction performance metrics—*hit rate* (h) and *response time* (t)—and lowers their average according to strength of COI. The second term can add up to half the hit rate to the R-value, depending on the strength of COI. Moderate reaction (R=0.5) is scored when two-thirds of the hit rate and reaction time is achieved with a COI of 0.5. Moderate or high R-values are always obtained when one of the variables is near maximum and the others are at least moderate.

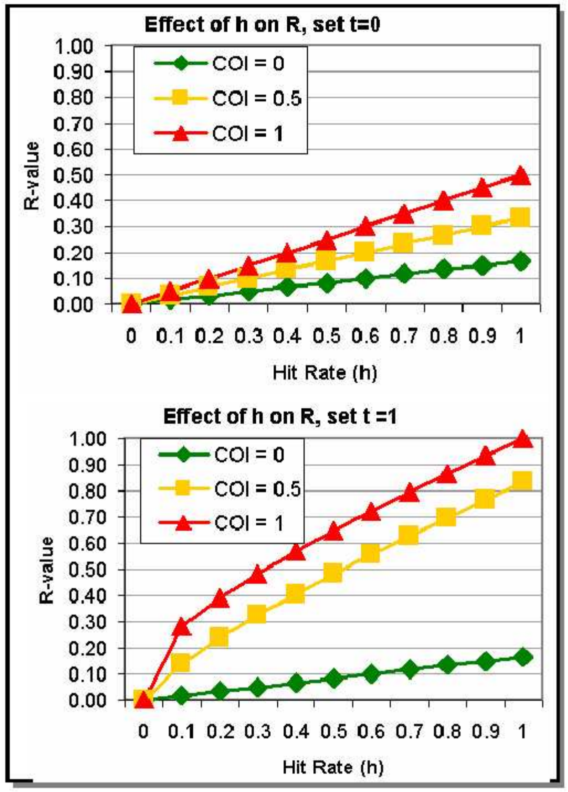
- The equation is also designed so that no more than R=0.5 can be achieved if one of the three variables equals zero. In order to understanding this rationale, one must considering that R is a characterization of an artifact's effectiveness for supporting reaction in a dual-task situation. That is, if the notification system is not attempting to resolve a situation constrained by the attention-utility theme (in which there would generally be at least a moderate value for COI), then the appeal of the artifact for facilitating notification reaction in a dual-task situation is inherently limited and therefore penalized. Both aspects of the reaction performance are also critical—a near-perfect hit rate would not be looked at as effective reaction if the response time is significantly slower than specification. Likewise, an acceptable response time has limited worth in the case that most signals delivered by the notification system are missed.

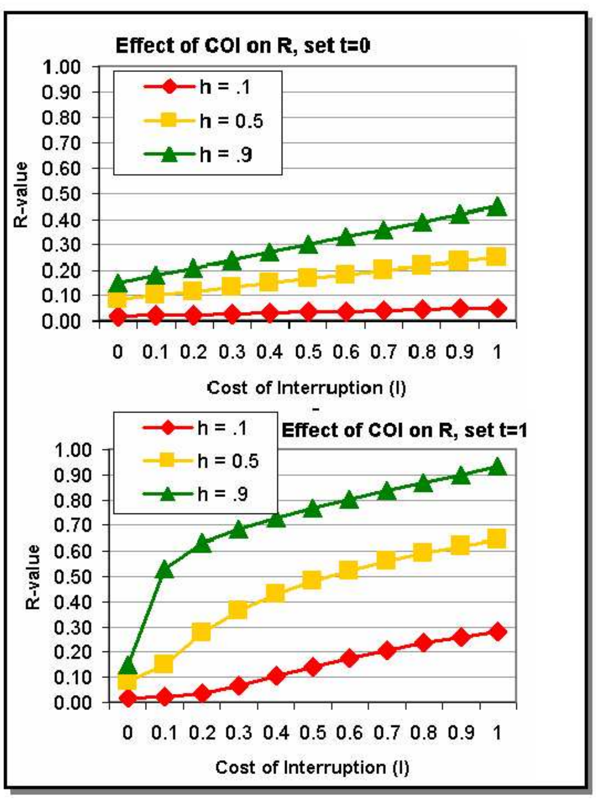
- Another feature of the equation is the prominence of the hit rate. Factoring this variable directly into both terms allows quick growth of R-values as hit rate increases, especially when COI is greater than 0.5. This adds a strong characteristic to R of being a measure of response selection probability. Since the test protocol relies on definition of total number of signals present, evaluators should ensure users are only expected to respond to a realistic number of important notifications. A suggested guideline for setting notification system test specifications is that expected reaction times and all signals can be hit at 100% without the primary task.

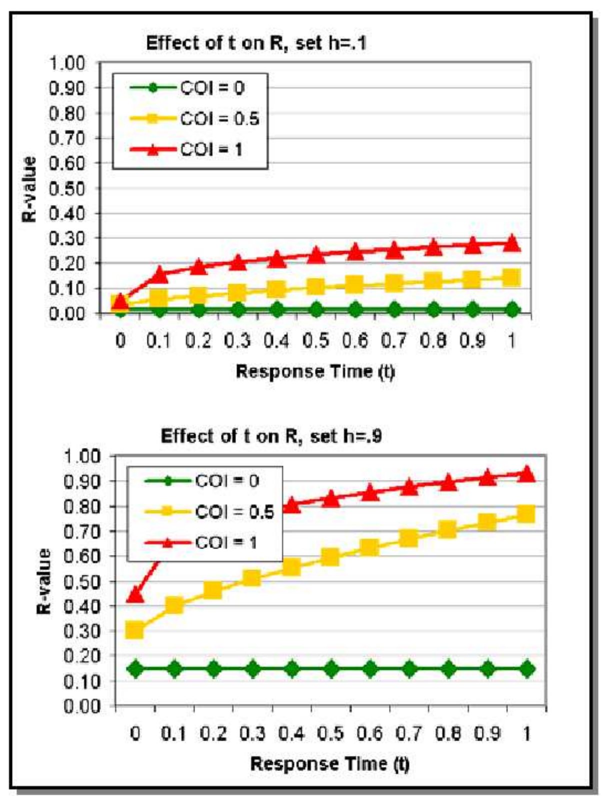
Sample results:

h	t	COI	R	Rationale
0.1	0	0	0.02	Certainly an **information design problem (perception)**—make notification more prominent
0.1	0	1	0.05	
0.1	0.5	0	0.02	Likely to be employing user initiated interruption or discrete attention division at inappropriate intervals—perhaps use an affordance to prompt monitoring cadence
0.1	0.5	1	0.23	Poor hit rate, but adequate response time in an urgent situation
0.3	0.5	0	0.05	
0.3	0.5	1	0.42	Decent response rate in an urgent situation, but too many signal are missed in this case
0.5	0	0	0.08	
0.5	0	1	0.25	
0.5	0.5	0	0.08	
0.5	0.5	0.5	0.37	Only moderate performance metrics in a moderately urgent situation—better than low R, but not by much
0.5	0.5	1	0.56	
0.7	0.5	0	0.12	
0.7	0.5	0.5	0.48	
0.7	0.5	1	0.70	
0.9	0	0	0.15	Hits most signals, but far too late and in a non-urgent situation. Most likely an **interaction design problem (pre-execution stage)**.
0.9	0	0.5	0.30	
0.9	0	1	0.45	Strong stimulus detection in an urgent situation, but extremely slow reaction time—likely an **interaction design problem (execution stage)**
0.9	0.5	0	0.15	
0.9	0.5	0.5	0.59	
0.9	0.5	1	0.83	
0.9	1	0	0.15	Although this artifact facilitates excellent reaction performance metrics, the non urgency of the dual-task situation prevents a strong R-value. Recommend retesting in a more urgent dual-task environment.
0.9	1	0.5	0.77	
0.9	1	1	0.93	

Figure B.2: Summarized justification and sample results for the R equation, *continued on the following page.*


Effect of h on R, set t=0
R-value
COI = 0
COI = 0.5
COI = 1
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Hit Rate (h)
Effect of h on R, set t =1
R-value
COI = 0
COI = 0.5
COI = 1
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Hit Rate (h)


Effect of COI on R, set t=0
R-value
h = .1
h = 0.5
h = .9
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Cost of Interruption (I)
Effect of COI on R, set t=1
R-value
h = .1
h = 0.5
h = .9
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Cost of Interruption (I)


Effect of t on R, set h=.1
R-value
COI = 0
COI = 0.5
COI = 1
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Response Time (t)
Effect of t on R, set h=.9
R-value
COI = 0
COI = 0.5
COI = 1
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Response Time (t)

$$\textbf{Comprehension: } \mathbf{C} = f + \frac{(1-f)(p+2c-cp)}{3}$$

Justification

- Based on concept of situation awareness, where a user accumulates Perception (of the elements in the system), Comprehension (of the current situation), and then Projection (of future status). Each level is dependent on achieving some part of the preceding level, and represents a progressively higher state of situated awareness [2]. Thinking of *C* as situation awareness brings it in line with a wealth of research in the human factors field. For instance, studies have shown that we can recognize the characteristics of awareness independent of the processes required to maintain it (working and long term memory or attentional state) [1] or the response selections that result from it [11].

- The simplified equation that appears above is difficult to explain, so we revert to the unsimplified version below:

$$\mathbf{C} = \frac{p+(1-p)(c+f(1-c))}{3} + \frac{c+f(1-c)}{3} + \frac{f}{3}$$

This equation consists of three terms—one for each level of situation awareness. As each level is maximized, the equation ensures that C=0.33, 0.67, and 1 respectively. If a given level is not maximized, achievement in the higher levels provide credit toward the C-value. This scoring scheme relies on an experimental protocol that discourages participant guessing or uncertainty.

- Still under review is the issue of whether COI should be an additional factor in the C equation. Some justification for this is present in Endsley's argument that temporal dynamics play an important part in assessing the comprehension and projection levels. Specifically, she mentions that part of projection requires an understanding of the rate at which information is changing.

Sample results:

f	c	p	C	Rationale
0	0	0	0	
0	0	0.5	0.17	
0	0	1	0.33	Full perception, with no interpretation—no long term awareness or relating notification content to user goals
0	.5	0	0.33	
0	.5	0.5	0.42	
0	.5	1	0.50	
0	1	d*	0.67	Full comprehension of single elements, but no ability to connect elements together to project a trend
.5	.5	0	0.67	An unlikely case that should be prohibited by experimental protocol
.5	.5	0.5	0.71	
.5	.5	1	0.75	
.5	1	d	0.83	c=0 and/or p=0 should be prohibited by experimental protocol
1	d	d	1	

* d = "don't care": any value produces same result

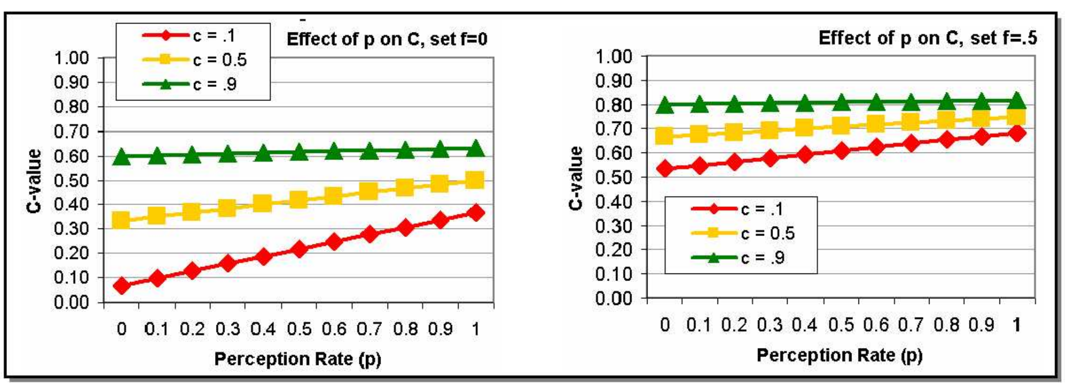


Figure B.3: Summarized justification and sample results for the C equation.

B.2 Experiment Materials

Test Script: IRC user's model (analytical method)

1. Have participants sign informed consent forms.

2. General instructions:

In this experiment, I want you to think of yourselves as expert evaluators for interfaces like the Scope notification system. You are uniquely qualified to do this because you've been thinking about the design challenges throughout the semester. In this session, you will be evaluating one the new Scope prototypes that was designed by students in your class. You will receive extra credit for participating in the study.

As we discuss a few specific aspects of the interface, you will be filling out the form in front of you. None of the responses that you provide are right or wrong, and they will be kept anonymous—I am considering your opinion to be an expert opinion, based on the experience you've acquired. None of the responses that you provide will have any impact at all on the grading of this prototype or any other prototype. In other words, if you chose to be critical and disagree or if you chose to be generous and agree, it will make no difference in the grades awarded by the TA and instructor. As you know, these prototypes were developed by their designers to test specific features and claims, which may not be the same aspects we will be considering today.

Please provide honest responses that reflect your expertise. If you have questions or would like to discuss aspects of the design with the facilitator or group as we go along, please do so. Please do not work ahead—wait for the overview before answering each question.

3. Prototype demo: [start appropriate prototype]

First, let's look at the system you will be evaluating. As stated by the design team's report, this new interface is intended to be…

[Group A] a ***bulletin board metaphor*** *that increases visual complexity, but will allow users to see at a glance what alerts need attending.*

[Group B] this ***task-bar-like*** *with several different divisions that split the bar into quarters, but allows the user to gain back screen space while unobtrusively receiving notifications.*

[Group C] a ***ladder metaphor for a to-do list*** *that helps users recognize which tasks are most urgent.*

This is the system that you will be evaluating. I want you to use the prototype that you see to envision what the actual system would be like, and rate that system. To accomplish this quickly, we'll discuss relevant aspects of the system according to each question.

If one of the designers of the system is present, they are encouraged to provide additional information about the design. However, let's keep discussion of each question to less than 2 minutes so that we can complete the study in a timely manner.

Q1:

We can assume that the display will be used with typical desktop primary tasks, such as word processing, Internet browsing, or programming. Let's look at what happens when a new notification arrives that might interrupt that primary task.

Figure B.4: *Script used for the analytical IRCresults procedure, continued on the following pages.*

Designer, what are the typical animation techniques, color changes, audio chimes that would be used when a new notification arrives?

Group, if the user did not want to lose concentration in their primary task, how easy would these information design techniques be to ignore?

Keeping these points in mind, indicate your answer for question 1.

Q2:

To answer the second question, let's consider what it takes to detect and interpret a notification.

Designer, what are the typical steps a user would take to detect, interpret, understand, and act on a notification, before resuming the primary task? How long would that typically take?

Keeping that in mind, think about primary task performance according to some metric like average words per minute in typing a document. Estimate a user's typical rate when they are only focused on the primary task. Now think about how much that rate changes when the notification is present, and indicate your answer for question 2.

Q3:

Keeping in mind the information design techniques used to introduce new notifications, let's think about what portion of important changes the user will detect.

Designer, what would be an example of an "important change" that a user would want to notice? How frequently would you expect that to happen?

Group, how noticeable would these notifications be over the period of a few hours? Keeping these points in mind, indicate your answer for question 3.

Q4:

Designer, would your system ever suggest actions for users to take? What types of actions? How quickly do you expect that users will want to take these actions after delivery of a notification—a matter of seconds, minutes, hours?

Group, based on how noticeable the notifications are, and whether the user will develop a tendency to ignore new notifications, what will the actual reaction time be like?

Question 4 asks you to compare the expected time with the actual time—indicate your answer now.

Q5:

Let's think about how often a user would typically interact with this system in a 5 minute period of time. By interact, I mean a brief glance, a mouseover or click, or any other usage action.

Designer, beyond detecting important notifications, what other frequent interactions do you expect users would have with this system? How long would these take?

Question 5 asks you to estimate the ratio of interactions that happen in response to an important notification to all interactions—indicate your response now.

Q6:

Imagine that a user stands up from their computer and walks away, perhaps to go to a meeting or to lunch, after using this system for several hours. If they were waiting for a bus ten minutes later, let's think about what information in the interface they would want to remember or reflect back on.

Group, what parts of the information being presented would be valuable—assuming no limit to a person's memory and perfect information presentation? Indicate your own opinion for question 6.

Q7:

Now, let's think about how much of the total information would realistically be actually remembered?

Q8:

Sometimes, after using a system for an extended period of time, users can notice patterns or trends, and predict future notifications.

Designer, are there any features in this system that help user predict notification patterns?

Indicate your response to question 8, evaluating the success of these features.

---

4. Wrap up [collect response sheets]

On the back of this questionnaire, please note any questions that you had difficulty answering or choices you found not descriptive enough. Also, if you feel like too many of your responses were guesses and that your responses should not be considered to be expert responses, please make a note to that effect.

Thank you for your participation—you are free to go.

Answer these eight (8) questions about the feature you are evaluating:

1. Given the nature and importance of the user's primary task at the receipt of the notification, ***how costly would an interruption be?***

- ☐ **Extremely costly** – an interruption would have dire consequences because attention was diverted from a critical primary task
- ☐ **Very costly**
- ☐ **Moderately costly** – while the user may be bothered by an interruption, impact on the primary task is equally recoverable
- ☐ **Not very costly**
- ☐ **Not at all costly** – the user would not care if they were interrupted, or they could easily resume their primary task

2. Compared to the primary task performance before the notification delivery, ***how much does the primary task performance reduce*** *when the notification is present?*

- ☐ **No performance reduction** – users can monitor and interpret notifications and fully maintain primary task performance
- ☐ **Very little** primary task performance is lost
- ☐ **About half** of the primary task performance is lost (perhaps twice as slow or half the quality)
- ☐ **More than half** of primary task performance is lost
- ☐ **All** primary task performance completely stops, or quality is completely lost

3. ***How often will users actually notice important changes*** *in the notification system, as opposed to not noticing them?*

- ☐ **Always** – an important change would not be undetected
- ☐ **More than half**
- ☐ **About half** of all important changes would be detected
- ☐ **Less than half**
- ☐ **Never** – no important changes would be detected

4. In cases where a notification suggests an action for a user to take, ***how does the user's response time compare to the reasonably desired response time?***

- ☐ **Better response time,** as good as expected or no action required
- ☐ **Slightly slower** response time
- ☐ **About twice as slow** as expected
- ☐ **Much slower**
- ☐ **Extremely slow**, or action never taken

Figure B.5: Form used by the expert participants in the experiment, continued on the next page.

*5. When considering the total number of times a user interacts with the notification system, what is the **ratio of the interactions in response to an important notification vs. total interactions** (including those when no actual notification was being delivered, i.e., user checking on their own or thinking there was a notification)?*

- **1 to 1**
- **2 to 3**
- **1 to 2**
- **1 to 4**
- **More than 1 to 4**

*6. **How much of the notification content** will the user **want to remember** for more than 10 seconds after its presentation?*

- **All content** will be worth remembering
- **More than half**
- **About half**
- **Less than half**
- **None at all** – users will not want to remember the content, although it may prompt instantaneous reactions

*7. **How much of the notification content** will the user **actually remember** for more than 10 seconds after its presentation?*

- **All content will be remembered**
- **More than half**
- **About half**
- **Less than half**
- **None at all**

*8. Based on the notification content, **how successful will the user be** in making projections or predictions about future trends for the long-term state of the system being monitored?*

- **Extremely successful** – users will always be able to make accurate and insightful predictions about future states
- **Very successful**
- **Moderately successful**
- **Not very successful**
- **Not at all**, or supporting prediction is not a goal for this system

B.3 Analytical Experiment Data and Analysis

Designer ID	Team ID	Raw Data I-value	R-value	C-value	Team Mean I	R	C	Abs Diff. From Team Mean I	R	C	Abs Diff. From Overall Mean I	R	C
1	1	0.38	0.72	0.68	**0.43**	**0.72**	**0.63**	0.04	0.00	0.05	0.12	0.00	0.07
2	1	0.47	0.61	0.67				0.05	0.11	0.04	0.03	0.11	0.06
3	1	0.47	0.85	0.51				0.05	0.13	0.12	0.03	0.13	0.10
4	1	0.38	0.70	0.66				0.04	0.02	0.03	0.12	0.02	0.05
5	2	0.53	0.77	0.41	**0.49**	**0.74**	**0.49**	0.04	0.03	0.08	0.03	0.05	0.20
6	2	0.47	0.72	0.60				0.02	0.02	0.11	0.03	0.00	0.01
7	2	0.47	0.72	0.45				0.02	0.02	0.04	0.03	0.00	0.16
8	3	0.24	0.59	0.73	**0.64**	**0.70**	**0.70**	0.40	0.11	0.03	0.26	0.13	0.12
9	3	0.62	0.62	0.73				0.02	0.08	0.03	0.12	0.10	0.12
10	3	0.69	0.66	0.35				0.05	0.04	0.35	0.19	0.06	0.26
11	3	0.99	0.91	1.00				0.36	0.22	0.30	0.49	0.19	0.39
12	4	0.48	0.69	0.77	**0.48**	**0.63**	**0.65**	0.01	0.06	0.13	0.02	0.03	0.16
13	4	0.47	0.57	0.52				0.01	0.06	0.13	0.03	0.15	0.09
14	5	0.47	0.72	0.26	**0.47**	**0.72**	**0.40**	0.00	0.00	0.14	0.03	0.00	0.35
15	5	0.47	0.72	0.53				0.00	0.00	0.14	0.03	0.00	0.08
16	6	0.32	0.74	0.35	**0.48**	**0.76**	**0.54**	0.16	0.02	0.19	0.18	0.02	0.26
17	6	0.64	0.78	0.73				0.16	0.02	0.19	0.14	0.06	0.12
18	7	0.37	0.72	0.52	**0.37**	**0.72**	**0.52**	*n/a*			0.13	0.00	0.09
19	8	0.64	0.85	0.39	**0.62**	**0.81**	**0.52**	0.02	0.04	0.13	0.14	0.13	0.22
20	8	0.47	0.87	0.76				0.15	0.06	0.24	0.03	0.15	0.15
21	8	0.74	0.71	0.41				0.12	0.10	0.11	0.24	0.01	0.20
22	9	0.47	0.79	0.62	**0.63**	**0.76**	**0.49**	0.16	0.04	0.14	0.03	0.07	0.01
23	9	0.79	0.72	0.35				0.16	0.04	0.14	0.29	0.00	0.26
24	10	0.47	0.67	0.95	**0.47**	**0.72**	**0.89**	0.00	0.05	0.06	0.03	0.05	0.34
25	10	0.47	0.77	0.82				0.00	0.05	0.07	0.03	0.05	0.21
26	11	0.48	0.69	0.67	**0.55**	**0.78**	**0.78**	0.07	0.09	0.11	0.02	0.03	0.06
27	11	0.62	0.86	0.89				0.07	0.09	0.11	0.12	0.14	0.28
28	12	0.14	0.77	0.83	**0.32**	**0.66**	**0.75**	0.18	0.11	0.08	0.36	0.05	0.22
29	12	0.34	0.44	0.70				0.02	0.22	0.05	0.16	0.28	0.09
30	12	0.47	0.77	0.71				0.15	0.11	0.04	0.03	0.05	0.10
31	13	0.38	0.69	0.66	**0.38**	**0.69**	**0.66**	*n/a*			0.12	0.03	0.05
32	14	0.47	0.75	0.71	**0.50**	**0.73**	**0.55**	0.03	0.02	0.17	0.03	0.03	0.10
33	14	0.37	0.86	0.55				0.13	0.13	0.01	0.13	0.14	0.06
34	14	0.37	0.74	0.47				0.13	0.01	0.08	0.13	0.02	0.14
35	14	0.79	0.57	0.45				0.29	0.16	0.10	0.29	0.15	0.16
36	15	0.50	0.80	0.34	**0.51**	**0.71**	**0.61**	0.01	0.09	0.27	0.00	0.08	0.27
37	15	0.57	0.67	0.77				0.06	0.04	0.16	0.07	0.05	0.16
38	15	0.47	0.66	0.71				0.04	0.05	0.10	0.03	0.06	0.10
	Overall Mean	**0.50**	**0.72**	**0.61**				0.09	0.07	0.12			

Figure B.6: Raw data (IRC ratings), team means, and differences between individual scores and team means and the overall mean.

Kendall's correlation

		Odd ID	Even ID
G1	*I*	0.38	0.47
		0.47	0.38
	R	0.72	0.61
		0.85	0.7
	C	0.68	0.67
		0.51	0.66
G2	*I*	0.53	0.47
	R	0.77	0.72
	C	0.41	0.6
G3	*I*	0.24	0.62
		0.69	0.99
	R	0.59	0.62
		0.66	0.91
	C	0.73	0.73
		0.35	1
G4	*I*	0.48	0.47
	R	0.69	0.57
	C	0.77	0.52
G5	*I*	0.47	0.47
	R	0.72	0.72
	C	0.26	0.53
G6	*I*	0.32	0.64
	R	0.74	0.78
	C	0.35	0.73
G8	*I*	0.64	0.47
	R	0.85	0.87
	C	0.39	0.76
G9	*I*	0.47	0.79
	R	0.79	0.72
	C	0.62	0.35
G10	*I*	0.47	0.47
	R	0.67	0.77
	C	0.95	0.82
G11	*I*	0.48	0.62
	R	0.69	0.86
	C	0.67	0.89
G12	*I*	0.14	0.34
	R	0.77	0.44
	C	0.83	0.7
G14	*I*	0.47	0.37
		0.37	0.79
	R	0.75	0.86
		0.74	0.57
	C	0.71	0.55
		0.47	0.45
G15	I	0.5	0.57
	R	0.8	0.67
	C	0.34	0.77
		Tau:	0.18
		P-value:	0.081

		Odd ID	Even ID
G1	*I*	0.38	0.47
		0.47	0.38
	R	0.72	0.61
		0.85	0.7
	C	0.68	0.67
		0.51	0.66
G2	*I*	0.53	0.47
	R	0.77	0.72
	C	0.41	0.6
G4	*I*	0.48	0.47
	R	0.69	0.57
	C	0.77	0.52
G5	*I*	0.47	0.47
	R	0.72	0.72
	C	0.26	0.53
G6	*I*	0.32	0.64
	R	0.74	0.78
	C	0.35	0.73
G8	*I*	0.64	0.47
	R	0.85	0.87
	C	0.39	0.76
G9	*I*	0.47	0.79
	R	0.79	0.72
	C	0.62	0.35
G10	*I*	0.47	0.47
	R	0.67	0.77
	C	0.95	0.82
G11	*I*	0.48	0.62
	R	0.69	0.86
	C	0.67	0.89
G12	*I*	0.14	0.34
	R	0.77	0.44
	C	0.83	0.7
G14	*I*	0.47	0.37
		0.37	0.79
	R	0.75	0.86
		0.74	0.57
	C	0.71	0.55
		0.47	0.45
G15	*I*	0.5	0.57
	R	0.8	0.67
	C	0.34	0.77
		Tau:	0.23
		P-value:	0.039
		(without Group 3)	

Figure B.7: Interrater reliabilities, showing tendency of an individual to agree with their team's mean rating. The second test was conducted without Team 3's results (this team was an outlier in terms of internal consistency).

Interface B

ID	Rating I	Rating R	Rating C	Diff from mean (rater) I	R	C	total
B1	0.88	0.59	0.16	0.53	0.29	0.45	1.27
B2	0.26	0.20	0.56	0.09	0.10	0.05	0.25
B3	0.16	0.09	0.42	0.19	0.21	0.19	0.60
B4	0.26	0.34	0.77	0.09	0.04	0.16	0.29
B5	0.26	0.14	0.64	0.09	0.16	0.03	0.28
B6	0.26	0.40	0.76	0.09	0.10	0.15	0.34
B7	0.35	0.43	0.82	0.00	0.13	0.21	0.34
B8	0.77	0.31	0.56	0.42	0.01	0.05	0.48
B9	0.26	0.34	0.45	0.09	0.04	0.16	0.29
B10	0.26	0.20	0.58	0.09	0.10	0.03	0.23
B11	0.26	0.40	0.83	0.09	0.10	0.22	0.41
B12	0.26	0.17	0.79	0.09	0.13	0.18	0.40

Initial Mean			Mean difference (group)			
0.35	0.30	0.61	0.16	0.12	0.16	0.43

Kendall's coefficient -		
	Tau:	0.57
	P-value:	0.004

[Remove outliers]

*outlier = total difference greater than .9 or any parameter over .4

ID	Rating I	Rating R	Rating C	Diff from mean I	R	C	Diff from overall I (*0.35*)	R (*0.37*)	C (*0.54*)
B2	0.26	0.20	0.56	0.00	0.07	0.10	0.09	0.17	0.02
B3	0.16	0.09	0.42	0.10	0.18	0.24	0.19	0.28	0.12
B4	0.26	0.34	0.77	0.00	0.07	0.11	0.09	0.03	0.23
B5	0.26	0.14	0.64	0.00	0.13	0.02	0.09	0.23	0.10
B6	0.26	0.40	0.76	0.00	0.13	0.10	0.09	0.03	0.22
B7	0.35	0.43	0.82	0.09	0.16	0.16	0.00	0.06	0.28
B9	0.26	0.34	0.45	0.00	0.07	0.21	0.09	0.03	0.09
B10	0.26	0.20	0.58	0.00	0.07	0.08	0.09	0.17	0.04
B11	0.26	0.40	0.83	0.00	0.13	0.17	0.09	0.03	0.29
B12	0.26	0.17	0.79	0.00	0.10	0.13	0.09	0.20	0.25

Adjusted Mean			Mean difference		
0.26	**0.27**	**0.66**	0.02	0.11	0.13

Figure B.8: Raw data (IRC ratings) for Interface B, rating group means, and differences between individual ratings and team means and the overall mean of all three interfaces.

Interface A

ID	Rating I	Rating R	Rating C	Diff from mean (rater) I	R	C	total
A1	0.35	0.66	0.65	0.03	0.24	0.27	0.54
A2	0.16	0.23	0.04	0.16	0.19	0.34	0.69
A3	0.35	0.48	0.46	0.03	0.06	0.08	0.17
A4	0.35	0.48	0.76	0.03	0.06	0.38	0.47
A5	0.00	0.20	0.13	0.32	0.22	0.25	0.79
A6	0.35	0.59	0.78	0.03	0.17	0.40	0.60
A7	0.35	0.48	0.31	0.03	0.06	0.07	0.15
A8	0.35	0.48	0.47	0.03	0.06	0.09	0.18
A9	0.26	0.34	0.20	0.06	0.08	0.18	0.32
A10	0.52	0.21	0.12	0.20	0.21	0.26	0.67
A11	0.52	0.50	0.23	0.20	0.08	0.15	0.42

Initial Mean

I	R	C
0.32	0.42	0.38

Mean difference (group)

I	R	C	total
0.10	0.13	0.22	0.45

[Separate by comprehension opinion]

ID	Rating I	R	C	Diff from mean I	R	C	Diff from overall I (*0.35*)	R (*0.37*)	C (*0.54*)
A1	0.35	0.66	0.65	0.00	0.12	0.03	0.00	0.29	0.11
A3	0.35	0.48	0.46	0.00	0.06	0.16	0.00	0.11	0.08
A4	0.35	0.48	0.76	0.00	0.06	0.14	0.00	0.11	0.22
A6	0.35	0.59	0.78	0.00	0.05	0.16	0.00	0.22	0.24
A8	0.35	0.48	0.47	0.00	0.06	0.15	0.00	0.11	0.07

Adjusted Mean

I	R	C
0.35	0.54	0.62

Mean difference

I	R	C
0.00	0.07	0.13

ID	Rating I	R	C	Diff from mean I	R	C	Diff from overall I (*0.35*)	R (*0.37*)	C (*0.54*)
A2	0.16	0.23	0.04	0.14	0.10	0.13	0.19	0.14	0.50
A5	0.00	0.20	0.13	0.30	0.13	0.04	0.35	0.17	0.41
A7	0.35	0.48	0.31	0.05	0.15	0.14	0.00	0.11	0.23
A9	0.26	0.34	0.20	0.04	0.01	0.03	0.09	0.03	0.34
A10	0.52	0.21	0.12	0.22	0.12	0.05	0.17	0.16	0.42
A11	0.52	0.50	0.23	0.22	0.17	0.06	0.17	0.13	0.31

Adjusted Mean

I	R	C
0.30	0.33	0.17

Mean difference

I	R	C
0.16	0.11	0.08

Kendall's coefficient -

Tau:	0.6
P-value:	0.002

Figure B.9: Raw data (IRC ratings) for Interface A, rating group means, and differences between individual ratings and team means and the overall mean of all three interfaces.

Interface C

ID	Rating I	Rating R	Rating C	Diff from mean (rater) I	R	C	total
C1	1.00	0.21	0.38	0.48	0.02	0.14	0.64
C2	0.35	0.25	0.64	0.17	0.02	0.12	0.31
C3	0.26	0.11	0.56	0.26	0.12	0.04	0.42
C4	0.16	0.09	0.68	0.36	0.14	0.16	0.66
C5	0.35	0.43	0.70	0.17	0.20	0.18	0.55
C6	0.65	0.25	0.58	0.13	0.02	0.06	0.22
C7	0.65	0.21	0.42	0.13	0.02	0.10	0.25
C8	0.77	0.14	0.37	0.25	0.09	0.15	0.49
C9	0.88	0.25	0.48	0.36	0.02	0.04	0.42
C10	0.35	0.21	0.23	0.17	0.02	0.29	0.47
C11	0.26	0.34	0.63	0.26	0.11	0.11	0.48

Initial Mean

0.52	0.23	0.52

Mean difference (group)

0.25	0.07	0.13

Kendall's coefficient - Tau: 0.41 P-value: 0.040

[Separate by interruption opinion]

	Rating I	R	C	Diff from mean I	R	C	Diff from overall I (*0.35*)	R (*0.37*)	C (*0.54*)
C2	0.35	0.25	0.64	0.06	0.01	0.07	0.00	0.12	0.10
C3	0.26	0.11	0.56	0.03	0.13	0.01	0.09	0.26	0.02
C4	0.16	0.09	0.68	0.13	0.15	0.11	0.19	0.28	0.14
C5	0.35	0.43	0.70	0.06	0.19	0.13	0.00	0.06	0.16
C10	0.35	0.21	0.23	0.06	0.03	0.34	0.00	0.16	0.31
C11	0.26	0.34	0.63	0.03	0.10	0.06	0.09	0.03	0.09

Adjusted Mean: 0.29 0.24 0.57 — **Mean difference**: 0.06 0.10 0.12

	Rating I	R	C	Diff from mean I	R	C	Diff from overall I (*0.35*)	R (*0.37*)	C (*0.54*)
C1	1.00	0.21	0.38	0.21	0.00	0.07	0.65	0.16	0.16
C6	0.65	0.25	0.58	0.14	0.04	0.13	0.30	0.12	0.04
C7	0.65	0.21	0.42	0.14	0.00	0.03	0.30	0.16	0.12
C8	0.77	0.14	0.37	0.02	0.07	0.08	0.42	0.23	0.17
C9	0.88	0.25	0.48	0.09	0.04	0.03	0.53	0.12	0.06

Adjusted Mean: 0.79 0.21 0.45 — **Mean difference**: 0.12 0.03 0.07

Figure B.10: Raw data (IRC ratings) for Interface C, rating group means, and differences between individual ratings and team means and the overall mean of all three interfaces.

B.4 Empirical Experiment Data and Analysis

C# Design Contest Project Specifications Fall 2003

Virginia Tech C# Design Contest

Fall 2003

General Instructions:

All contestants must address the design problem detailed below using only C#. For this project you may work individually or in groups. The problem is intentionally open-ended to provide you with room for creativity. Additional clarification and resources are available from the project web site at: http://ticker.cs.vt.edu/~csharp/. All teams entering should register by sending full contact details for each member to csharp.net@vt.edu. All submissions must be received by October 26th 11:59:59 p.m. as detailed in the deliverables section. Prizes will be given to the top three submissions with the first place winning team receiving $400, second place receiving $200, and third place receiving $100. Any questions about this set of specifications should also be directed to csharp.net@vt.edu.

Design Problem:

You have just been hired by a local company as part of a C# development team. This company flies various personnel between Virginia and Los Angeles on a regular basis, using tickets purchased on Expedia.com. Unfortunately, many workers spend hours surfing Expedia for the best deals on tickets; decreasing productivity. Your assignment is to build a desktop notification display that can monitor Expedia, while still allowing workers to do their jobs. Notification displays are usually persistent interfaces that run in the corner of a user's computer screen. Examples include ESPN's BottomLine, Weather Bug, or even pop-up reminders like those found in MSN Messenger. Links to examples can be found on the project web site at http://ticker.cs.vt.edu/~csharp/resources.html.

Knowing that this system will be used throughout the company, you design it around some basic guidelines:

- The desktop display should be relatively small (not so small as to cause annoyance), and always visible (in some form).
- The display should not be annoying or offensive to view even after extended periods of use.
- The system use must be entirely intuitive for typical college-aged computer users; all features should be understandable without accessing a help file (tool tips are allowed).

The company prefers that its workers depart from one of four airports in VA: Roanoke, Greensboro, Charlottesville, or Dulles; and arrive in California at either Los Angeles or Long Beach Airport. Your first prototype of this application is scheduled to be delivered to the company board for review by October 2003.

Figure B.11: Specification for the Expedia notification systems.

Prototype Testing:

Content from Expedia.com has been archived and made available on the project web site at: http://ticker.cs.vt.edu/~csharp/cache.html. This data was recorded over a two hour period. All teams will submit a prototype along with their deliverables which should model this data and present it to the user in a notification desktop display.

All prototypes will be run for a 10 minute simulation on a system running Windows XP, IE 5+, Visual Studio .NET, an active connection to the internet and 1024 x 768 resolution. While ticket information was compiled from Expedia.com over a 2 hour period, your program should condense this data into a 10 minute simulation as if it were actually monitoring the data in real-time. In other words, your prototype will reflect in ten minutes what the cached data online shows in 2 hours. How and what you choose to display will be the core of how your program will be judged.

Please note that since your prototype presents merely a simulation, it is not necessary that your application go out onto the internet and actually parse the data (in fact doing so will **not** earn you any additional merit during judging). Instead, consider the online data a script that your program should follow. Display this data in both a creative and helpful manner as if your program had monitored Expedia.com for two hours, but only had 10 minutes to re-create what it saw. Remember to make your system useful; you want people to use your application for acquiring airline tickets in the future.

Each design will be independently evaluated by multiple judges according to metrics that capture these stated specifications:

- **Persistent use of screen space** – judged on average total pixels during usage (is it too small to see or so large that it gets in the way of other computer activities?)
- **Annoyance factor** – relative number of times during evaluation period that a judge decides interface is “inappropriately distracting” or “annoying”
- **Usefulness** - what information is displayed (for instance, displaying what flights are available may or may not be more important than showing driving distances)
- **Aesthetic and creative quality** by which the information is presented
- **Learnability** – scored after three minutes of unguided exposure to the interface

The online data will remain posted during prototype testing, so if you choose to link from your prototype to the web pages, they will be available. You should adhere to good programming style throughout your application and make use of some of the more economizing features of C#. In fact, your application **must** employ two of the following features: delegates, reflection, or interfaces. In retrospect, you will be focusing on the interface aspect of the project as well as demonstrating your knowledge of C#. *Since the prototype is a notification system, it may be a good idea to make use of the PictureBox control to add graphics to your forms.*

Deliverables:

C# Design Contest Project Specifications Fall 2003

Each team will submit a .zip archive containing the following:

- A text file named "team.txt" with the contact information of all individuals responsible for the project. Any prizes awarded will be given to the individuals listed in this file to divide among them (again, single participants are allowed, and must still include this file in their archive).
- Your project directory containing all files necessary to rebuild your project in Visual Studio .NET (special case: if you employ any third party libraries or COM components in your project, submit the source code for these with your archive only pursuant with the distribution clause of your license agreement). Do **not** include the obj directory within your project directory; however, you **must** include the bin directory within your project directory, with a compiled executable of your program therein.
- A text file named "reqs.txt" which details which two of the three C# features were used, and the files and function definitions they can be found in. These examples should be well documented in your source code and will be reviewed by the judges. Only one example of each should be named in reqs.txt, regardless of how many times you may use these techniques throughout your project.
- A text file named "design.txt" which should include a description of the design model for your prototype detailing how you anticipate the interface will affect user interruption from ongoing tasks, reaction to Expedia information, and understanding of patterns and trends of ticket prices over time from the four locations. Also note any other important effects on users you anticipate.

Your archive should be named using a single PID and a revision number. For instance, if "csharp.net" was the PID of someone in your group, then your first archive might be named "csharp.net1.0.zip". Subsequent submissions should increment the revision number 1.0 accordingly. There is a comfortable limit of **ten** uploads per team enforced. Any submissions after ten will be disregarded. Instructions detailing the submission process with screen shots are available on the project web site at: http://ticker.cs.vt.edu/~csharp/howtosubmit.html.

Submit your archive to the project web site at: http://ticker.cs.vt.edu/~csharp/upload.php by October 26th at 11:59:59 p.m. **Any archive that does not adhere to the above criteria will be automatically disqualified.**

Cheapest flight price by time

Example: A flight that apears after 1 hour in real time should appear in the notification system at 5 minutes.

real time(over 2 hours)	0	12	13	15	16	19	24	30	31	32	34	35	39	40	41	42	43	61	62	63	64	67	71	72	75	76	77	83	88	94	95	96	100	101	102	104	105	107	111	120
simulated time(over 10 minutes	0	1	1.1	1.3	1.3	1.58	2	2.5	2.6	2.7	2.8	2.9	3.3	3.3	3.4	3.5	3.6	5.1	5.2	5.3	5.3	5.6	5.9	6	6.3	6.3	6.4	6.9	7.3	7.8	7.9	8	8.3	8.4	8.5	8.7	8.8	8.9	9.3	10
ROA-LAX	294					300				310						320				330							340			350				365						390
IAD-LAX	228	328						428						444								434		424									308		295					290
CHO-LAX	445	450							455					460						465			470					475										375		275
GSO-LAX	293			285						275						285			273							260					270				285					301
ROA-LGB	501				511		521									530			538								532		548									532		522
IAD-LGB	308				408						508				510			500							483						463								432	419
CHO-LGB	500		510									520					525		530					530					430								330			230
GSO-LGB	422		412							410			410								400						395					400				410				415

Figure B.12: Data provided to Expedia notification system designers, consisting of times at which ticket prices for eight different flights change.

LAX Flights

Criteria:
For prices that have been over $300, buy if they go less than $300
For prices that start under $300 or a previously purchased amount, buy a ticket if they start rising
Never buy a ticket that's more than you already paid for.

Key: = *Event (Times)* = *Signal (Price)*

Timeline

ROA-LAX	0	19	32	42	63	77	94	101	120
	294	300	310	320	330	340	350	365	390
	0:00	1.58	2.7	3.5	5.3	6.4	7.8	8.4	10
IAD-LAX	0	12	30	40	67	72	100	102	120
	228	328	428	444	434	424	308	295	290
	0	1	2.5	3.3	5.6	6	8.3	8.5	10
CHO-LAX	0	12	31	40	63	71	83	107	120
	445	450	455	460	465	470	475	375	275
	0	1	2.6	3.3	5.3	5.9	6.9	8.9	10
GSO-LAX	0	15	32	42	62	76	95	102	120
	293	285	275	285	273	260	270	285	301
	0	1.3	2.7	3.5	5.2	6.3	7.9	8.5	10

Signal Timings:

Signal	1|.5|.25
Signal	1.58|.5|.25
Signal	3.5|.5|.25
Signal	7.9|.5|.25
Signal	8.5|.5|.25

Comprehension-
How many of the LAX flight prices dropped lower than $300?
Which route had the most expensive ticket price?
Which flight would be best to purchase as the program cut off?
Which flight increased steeply in price near the end of the round?

Projection-
Which best describes the general price trend of the CHO-LAX flight?
Which LAX flight seems to have the most stable ticket prices?
Which LAX flight may have a tendency of becoming more expensive during this time period?

Figure B.13: Experiment setup criteria for the LAX flights, as well as resulting signal definitions, comprehension and projection questions.

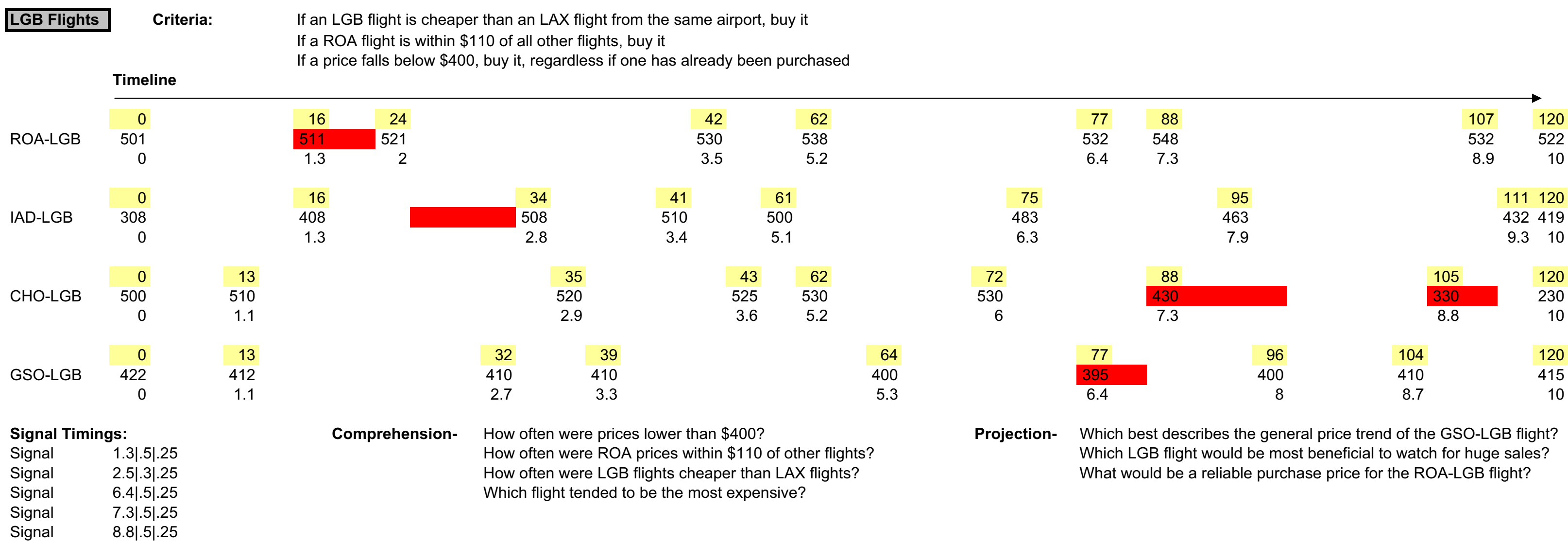


Figure B.14: Experiment setup criteria for the LGB flights, as well as resulting signal definitions, comprehension and projection questions.

COI: 0.33

Target IRC: I: 0.2, R: 0.44, C: 0.8

Particip ID	Ptask Benchmark	Interface	*Primary Task Performance* Pre-signal catches	Pre-signal total	In-signal catches	In-signal total	Dual-task Performance	Sustainment	*Notification Task Performance* Hits	False Alarms	Total Signals	Hit Rate	Perception Rate	Response Rate	Comprehension Correctness	Projection Correctness	I	R	C
1	0.84	**B**	301	457	105	138	0.68	**0.81**	1	5	5	**0.2**	**0.17**	**1.00**	**0.25**	**0.33**	0.19	0.15	0.47
2	0.74		306	451	115	141	0.71	**0.97**	1	1	5	**0.2**	**0.50**	**0.63**	**0.00**	**0.00**	0.03	0.12	0.17
3	0.75		235	438	74	157	0.52	**0.69**	0	2	5	**0**	**0.00**	**0.00**	**0.00**	**0.00**	0.31	0.00	0.00
4	0.70		309	461	102	134	0.69	**0.99**	1	8	5	**0.2**	**0.11**	**1.00**	**0.00**	**0.00**	0.01	0.15	0.04
5	0.78		308	471	96	121	0.68	**0.87**	1	1	5	**0.2**	**0.50**	**1.00**	**0.00**	**0.00**	0.13	0.15	0.17
6	0.88		356	492	70	102	0.72	**0.82**	2	1	5	**0.4**	**0.67**	**1.00**	**0.00**	**0.00**	0.18	0.31	0.22
																	0.14	0.15	0.18
7	0.82	**C**	325	463	91	132	0.70	**0.86**	2	7	5	**0.4**	**0.22**	**1.00**	**0.50**	**0.33**	0.14	0.31	0.58
8	0.70		240	431	93	164	0.56	**0.80**	0	2	5	**0**	**0.00**	**0.00**	**0.25**	**0.00**	0.20	0.00	0.17
9	0.70		285	441	109	153	0.66	**0.95**	1	7	5	**0.2**	**0.13**	**0.56**	**0.00**	**0.00**	0.05	0.11	0.04
10	0.76		232	437	106	158	0.57	**0.75**	0	1	5	**0**	**0.00**	**0.00**	**0.00**	**0.00**	0.25	0.00	0.00
11	0.86		319	442	93	151	0.69	**0.80**	2	2	5	**0.4**	**0.50**	**1.00**	**0.25**	**0.00**	0.19	0.31	0.29
12	0.90		363	459	109	134	0.80	**0.88**	1	5	5	**0.2**	**0.17**	**0.56**	**0.25**	**0.67**	0.12	0.11	0.74
																	0.16	0.14	0.30
13	0.84	**A**	343	444	106	148	0.76	**0.90**	0	2	5	**0**	**0.00**	**0.00**	**0.25**	**0.00**	0.10	0.00	0.17
14	0.76		204	439	78	156	0.47	**0.62**	1	7	5	**0.2**	**0.13**	**0.71**	**0.00**	**0.00**	0.38	0.13	0.04
15	0.82		314	495	72	98	0.65	**0.80**	4	12	5	**0.8**	**0.25**	**1.00**	**0.00**	**0.00**	0.20	0.62	0.08
16	0.82		289	469	91	126	0.64	**0.77**	1	7	5	**0.2**	**0.13**	**1.00**	**0.75**	**0.33**	0.22	0.15	0.67
17	0.75		231	435	102	161	0.56	**0.74**	0	4	5	**0**	**0.00**	**0.00**	**0.00**	**0.00**	0.25	0.00	0.00
18	0.88		365	450	113	144	0.80	**0.92**	0	4	5	**0**	**0.00**	**0.00**	**0.00**	**0.33**	0.08	0.00	0.33
																	0.21	0.15	0.22

Figure B.15: *Raw data and user's model IRC values obtained from the LAX round.*

Appendix C

Claims Library and LINK-UP Materials

C.1 Claims Library Screenshots and Documentation

The claims library database is available at http://research.cs.vt.edu/ns/research/Christa/claimsDB.sql. This database includes real claims (claim IDs 8 to 53 and greater than 152182) as well as the 150,000+ automatically generated claims. The database tables include other information used to implement the LINK-UP system, but the “claim” table and immediate relations store the claims.

Field	Description
1. Claim ID	Unique identifier for the claim
2. Title	Description given by the claims author
3a. Author	Researcher(s) or practitioner(s) who developed the original claim
3b. Editor	Person who created the claim record for the library and established the IRC rating
4. Artifact (Design Abstraction)	Description of the claim artifact in terms of the design abstraction
5. Description	Natural-language description of the claim
6. Upsides	Positive effect of using claims on a management objective of system effectiveness
7. Downsides	Negative effects of using claims
8. Scenarios	Originating scenario in which the claim was derived
9. Effect (IRC Rating)	Desired, measurable effect that the implemented claim should achieve
10. Dependencies	Problems that need to be solved before getting to the root issue
11. Issues	Management issues possible influenced by the claim
12. Theory	Underlying theory explicitly reference by the claim
13. Relationship	Interclaim links that describe how claims evolved during the history of the investigation
14a. Scope (Primary Task)	Generalized Task keyword description
14b. Scope (Notification Task)	Generic Task keyword description
15. Parent Component	System name and information
16. Usage Log	Running commentary on experience in using the claim

Figure C.1: Modified claim format, where the first 14 fields correspond largely to Sutcliffe (2002).

Radar Metaphor for Content Classification

Rating

Description

Contents of multiple information resources are organized and displayed as items within a bulls-eye circle?colored quadrants reflect item type, while distance to center depicts item urgency.

Upsides

+ Position centering effectively supports glanceable judgment of item urgency
+ Radar metaphor is easily understood by a wide variety of users
+ Bulleye layout naturally directs user focus to the most important area of the display (the center, which holds high urgency items)
+ Different colored quadrants aid distinction of item categories

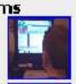
Downsides

- Item distance from center (relative urgency) is difficult to compare for items in different quadrants
- Most important items are clustered in the center, providing least screen space for display of icons
- Sharply contrasting quadrant colors may be distracting or provide poor contrast to icons

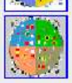
Scenario

As John works on his document and glances at the Scope interface, he is able to quickly discern whether emails, tasks, and calendar appointments are important or not. Currently, there are no icons in the center of the Scope, since he cleared out all of the urgent items earlier. He notices over the course of a few hours that several less important items have been added, and appreciates this overview of his inbox and appointment book (he can tell what types of items they are based on the quadrant they appear in). He can also see that one task is slowly moving toward the center, and he realizes that the deadline is approaching. As he is finalizing some parts in his document and preparing to switch to the task, he notices the arrival of a new, highly urgent email and decides to check it right away.

Scenario Diagrams


John using scope while browsing email


Basic overview of unimportant items


Noticing new important alerts arriving

Notification Task(s)	Primary Task(s)	Design Abstraction(s)
Decide	Other	Color
Compare	Planning - Scheduling	Screen Space
Locate		Grouping
Model		Metaphor

IRC Parameters

0.3 / 0.8 / 0.6

Figure C.2: Prototype of the claim record display screen (continued on next page).

Parent Artifact

Claim Artifacts

- Scope

Design Issues

comprehensibility of icons, color selection for quadrants, competing use of animation techniques, scalability

Related Theory

see Proximity Compatibility principle (Wickens & Carlswell, 1995), Rankine displays (Vicente et al, 1996), Weber's Law (Cleveland, 1994)

Artifact Media

The scope displaying an email message.

A scope prototype minimized on the desktop.

A scope prototype expanded on the desktop.

A sample quadrant "expanded."

Other quadrant layout ideas.

The scope integrated into a watch.

User Rating

User Comments

CR_Shepard - *09/04/2003* - After implementing the radar metaphor in our latest project my group found that the color quadrant system can be intimidating for new users, especially if more than 4 colors are used. The learning curve for the system rises rapidly for each new color used, so we recommend other users maintain 4 or less, or choose not to rely on colors as a main source of information.

TD_Briggs - *09/04/2003* - The radar metaphor proved especially useful in our new email tracking program, however we did have some issues with distinguishing text on items that have moved towards the center of the screen (high priority.) The radar still proved useful yet we recommend others have some sort of "pop-up text" so that the center doesn't become too cluttered.

Testing Results

Unknown

Author

PN_Truitt (email)

Editor

Content Group

Claim Id: 19

Top ^

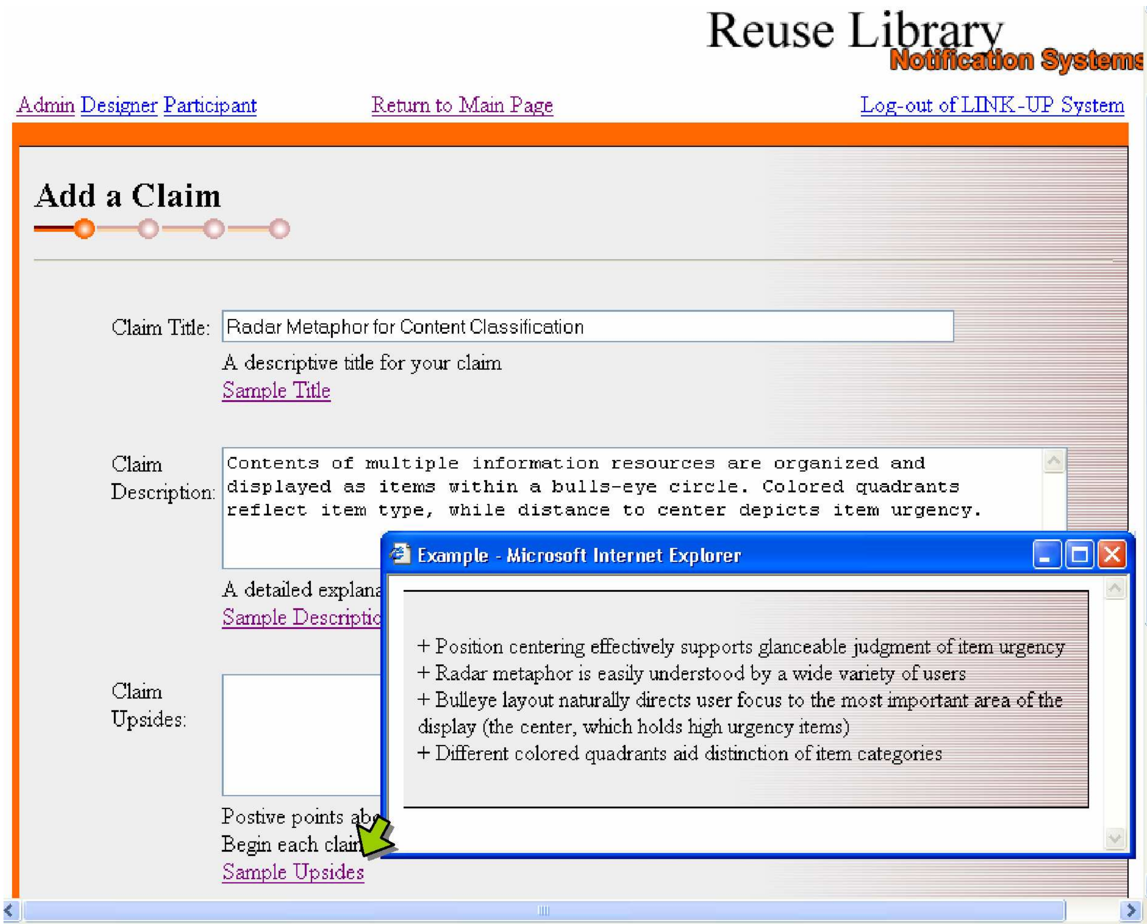


Figure C.3: Screenshot of the claim adding mechanism within the claims library. Four screens (like this one) allow the user to add all information about the claim, such as associated primary and notification tasks, IRC values, and multimedia objects that depict the artifact.

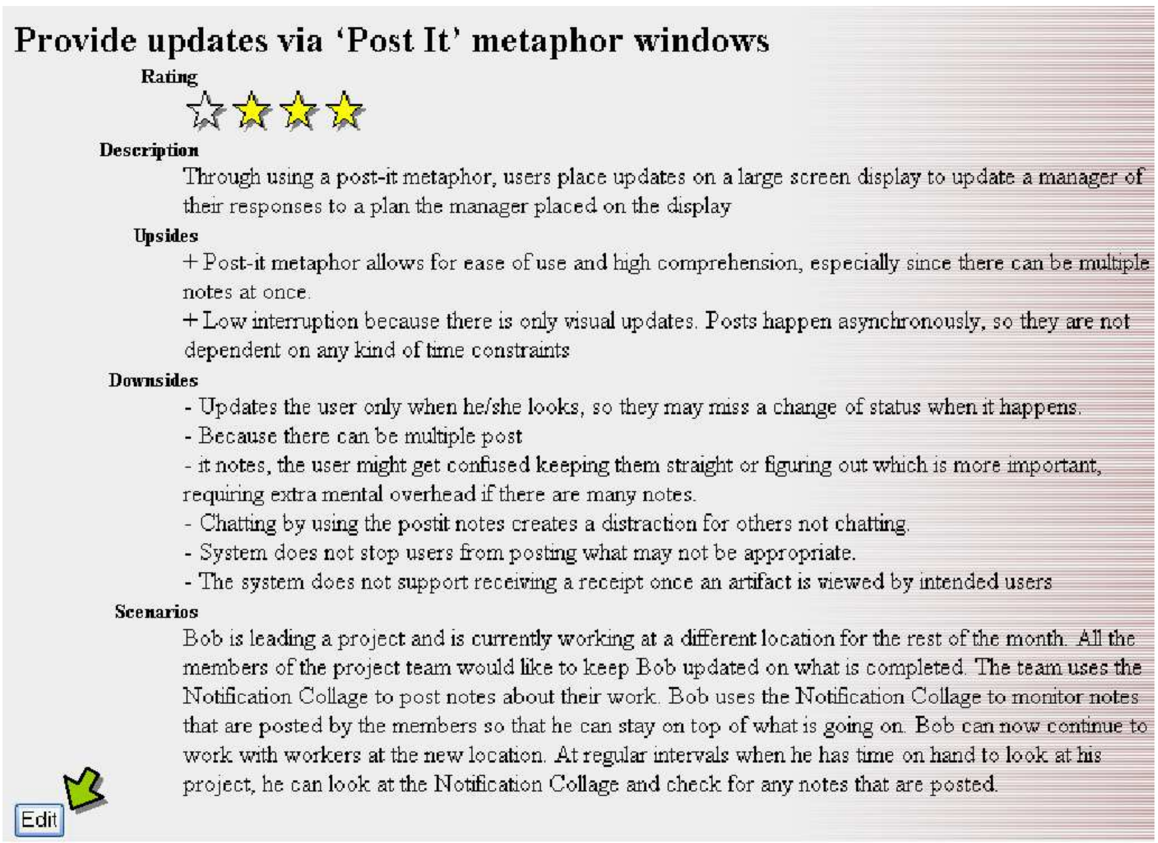
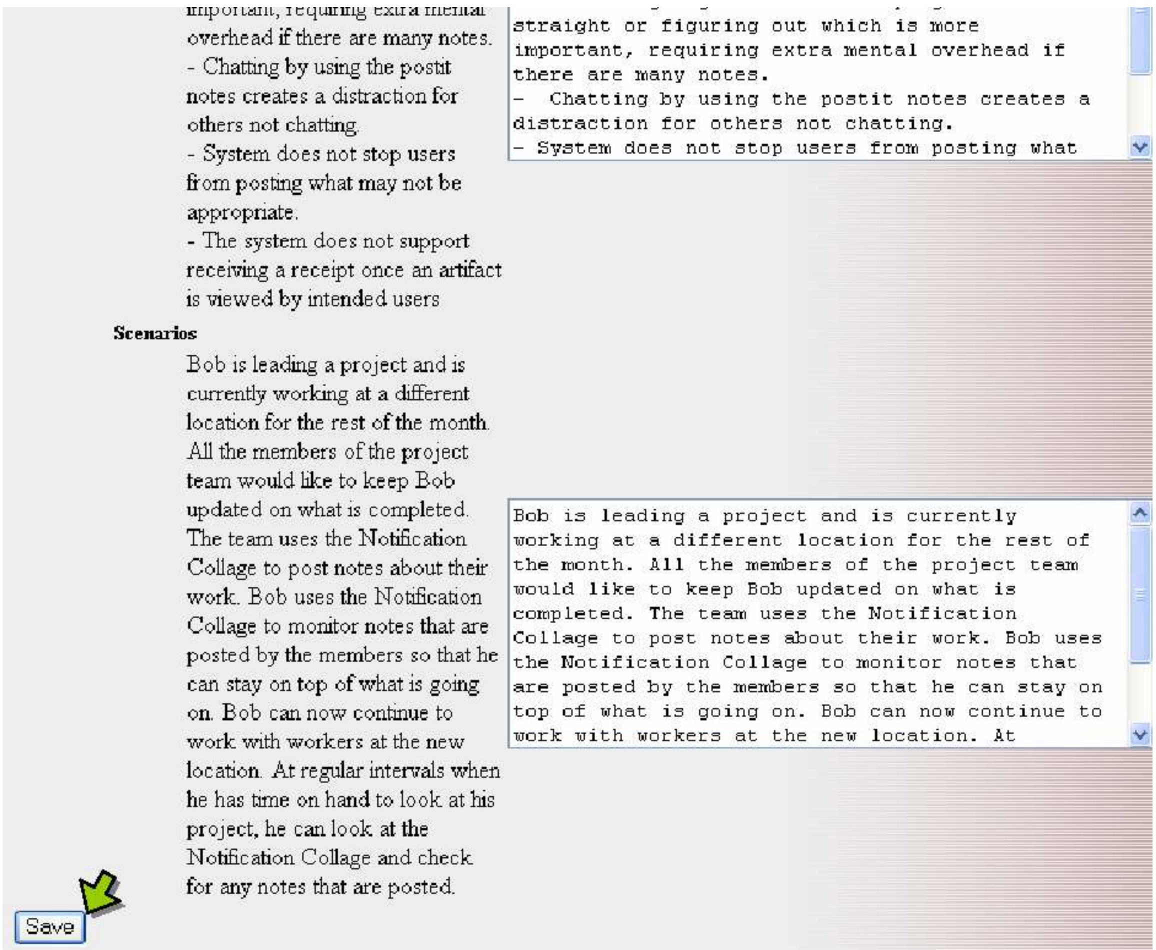


Figure C.4: When an owner of a claim opts to edit the claim already stored within the claims library, this screen appears. On this page, the user selects the portion of the claim record he want to change. The screen shown on the following page allows the editing operation to occur.



important, requiring extra mental overhead if there are many notes.
- Chatting by using the postit notes creates a distraction for others not chatting.
- System does not stop users from posting what may not be appropriate.
- The system does not support receiving a receipt once an artifact is viewed by intended users

straight or figuring out which is more important, requiring extra mental overhead if there are many notes.
- Chatting by using the postit notes creates a distraction for others not chatting.
- System does not stop users from posting what

Scenarios

Bob is leading a project and is currently working at a different location for the rest of the month. All the members of the project team would like to keep Bob updated on what is completed. The team uses the Notification Collage to post notes about their work. Bob uses the Notification Collage to monitor notes that are posted by the members so that he can stay on top of what is going on. Bob can now continue to work with workers at the new location. At regular intervals when he has time on hand to look at his project, he can look at the Notification Collage and check for any notes that are posted.

Bob is leading a project and is currently working at a different location for the rest of the month. All the members of the project team would like to keep Bob updated on what is completed. The team uses the Notification Collage to post notes about their work. Bob uses the Notification Collage to monitor notes that are posted by the members so that he can stay on top of what is going on. Bob can now continue to work with workers at the new location. At

Save

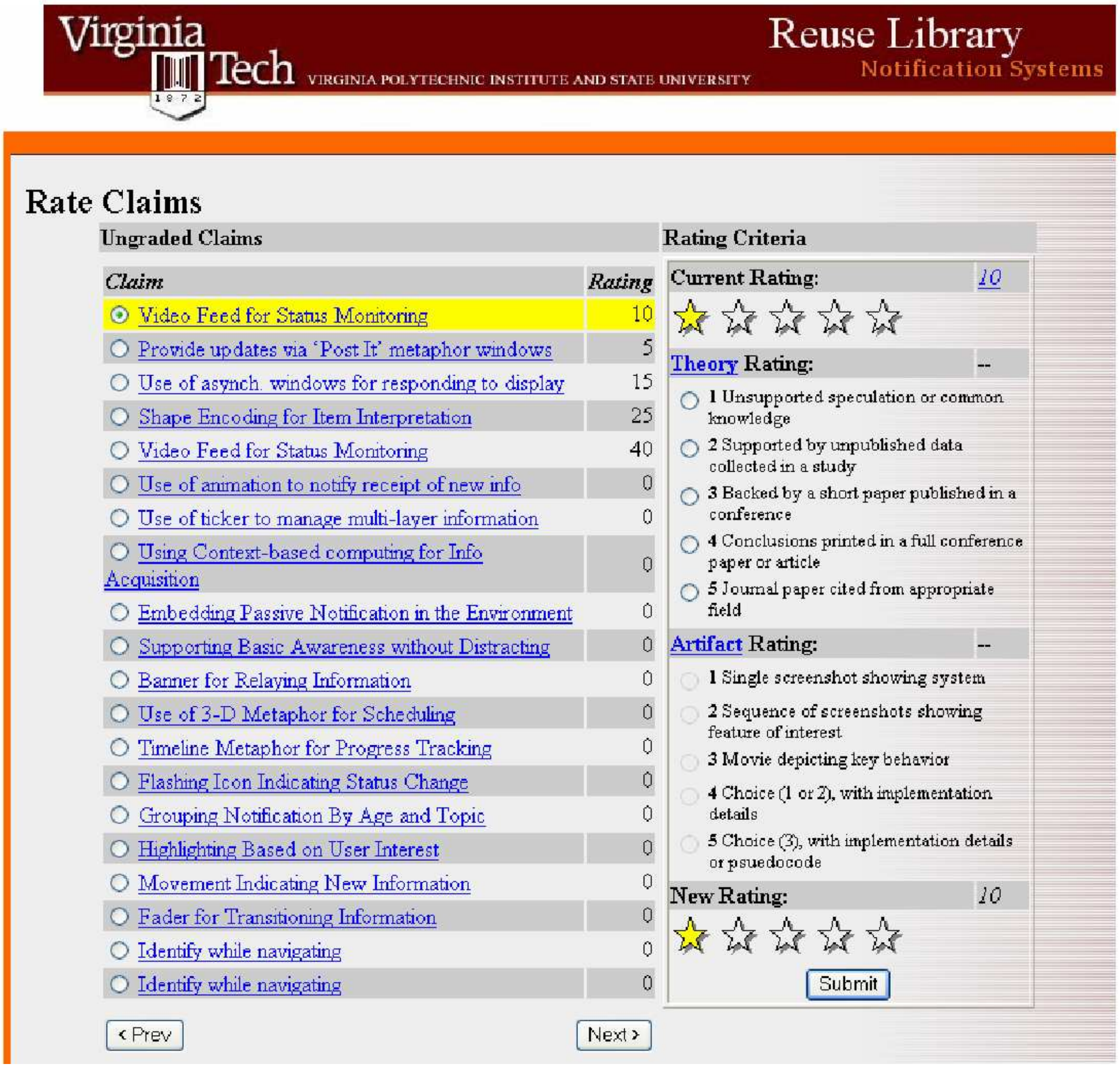


Figure C.5: Rating page for administrators to subjectively evaluate the quality of claims. All unrated claims appear in the left frame. The user is able to access information related to the other three rating factors (frequency of claim use in projects, average user rating, and claim author's experience level) with the link next to "Current Rating". Fields associated with the theory and artifact's multimedia components are accessible with respective links.

C.2 LINK-UP Screenshots and Pilot Test Results

C.2.1 Requirement Analysis Module screenshots

The following screens were developed and used for pilot testing by a team consisting of Dillon Bussert, Solomon Gifford, Cyril Montabert, and Melissa Grant. The team developed the module in JSP with Java servlets; technical assistance and server administration was provided by Edwin Bachetti. As with other design-process modules of LINK-UP, the project was specified and advised by Christa Chewar and Scott McCrickard.

Figure C.6: Screen within the Requirement Analysis Module, allowing a user to specify the design model IRC parameters (using IRCspec), as well as basic tasks include in the problem scenario. At this point in the interaction sequence, the user would have already created a project and entered problem scenarios.

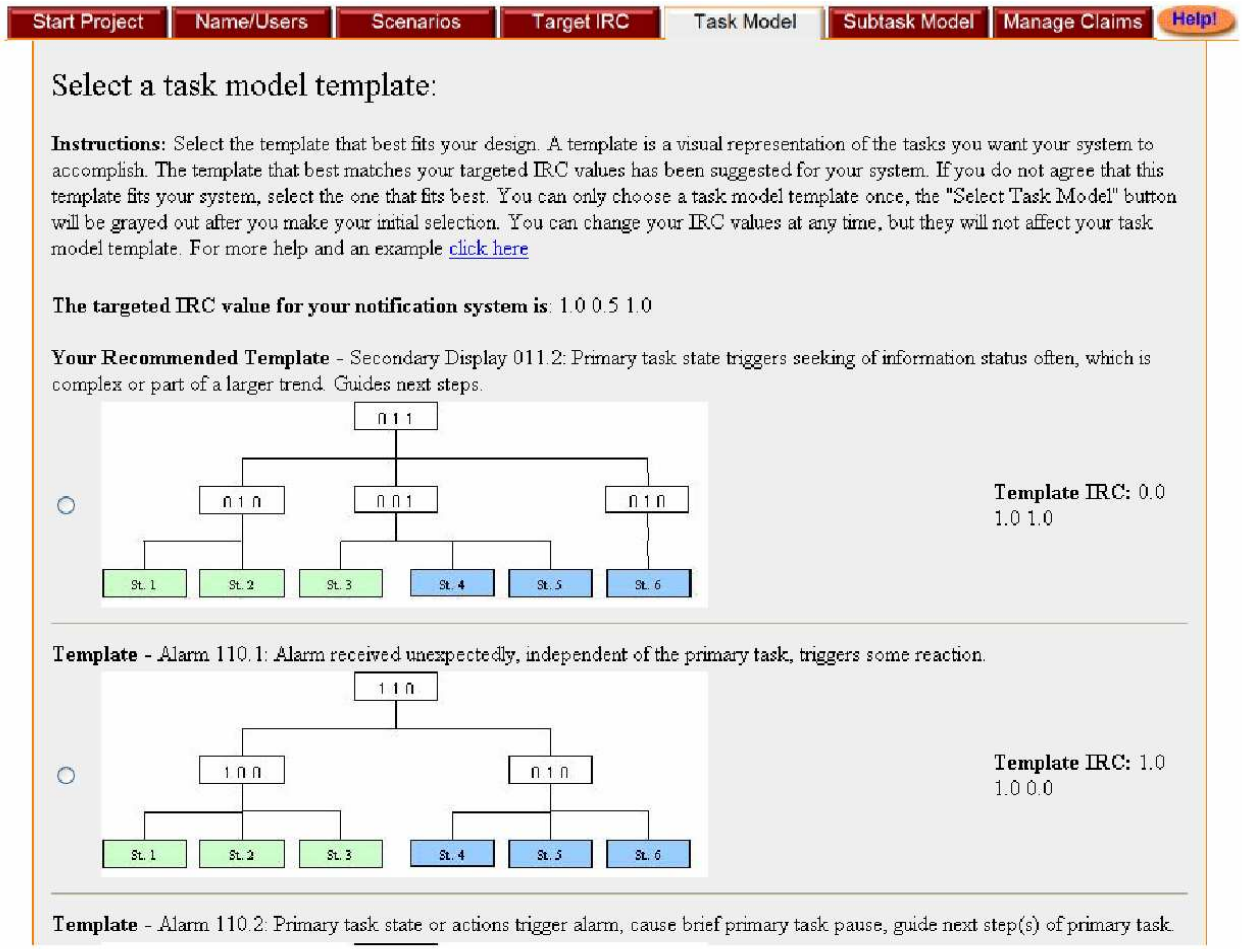


Figure C.7: After using the IRCspec tool to obtain design model IRC values for the a problem scenario, the designer is presented with recommendations for a generic task model. In the fully developed modules, many generic task models would be available, based on different combinations of IRC values and information processing strategies (i.e., top-down or bottom-up). Other information obtained from designers in the IRCspec interaction, as well as through the selection of basic tasks (generic and generalized) can also help to provide template recommendations.

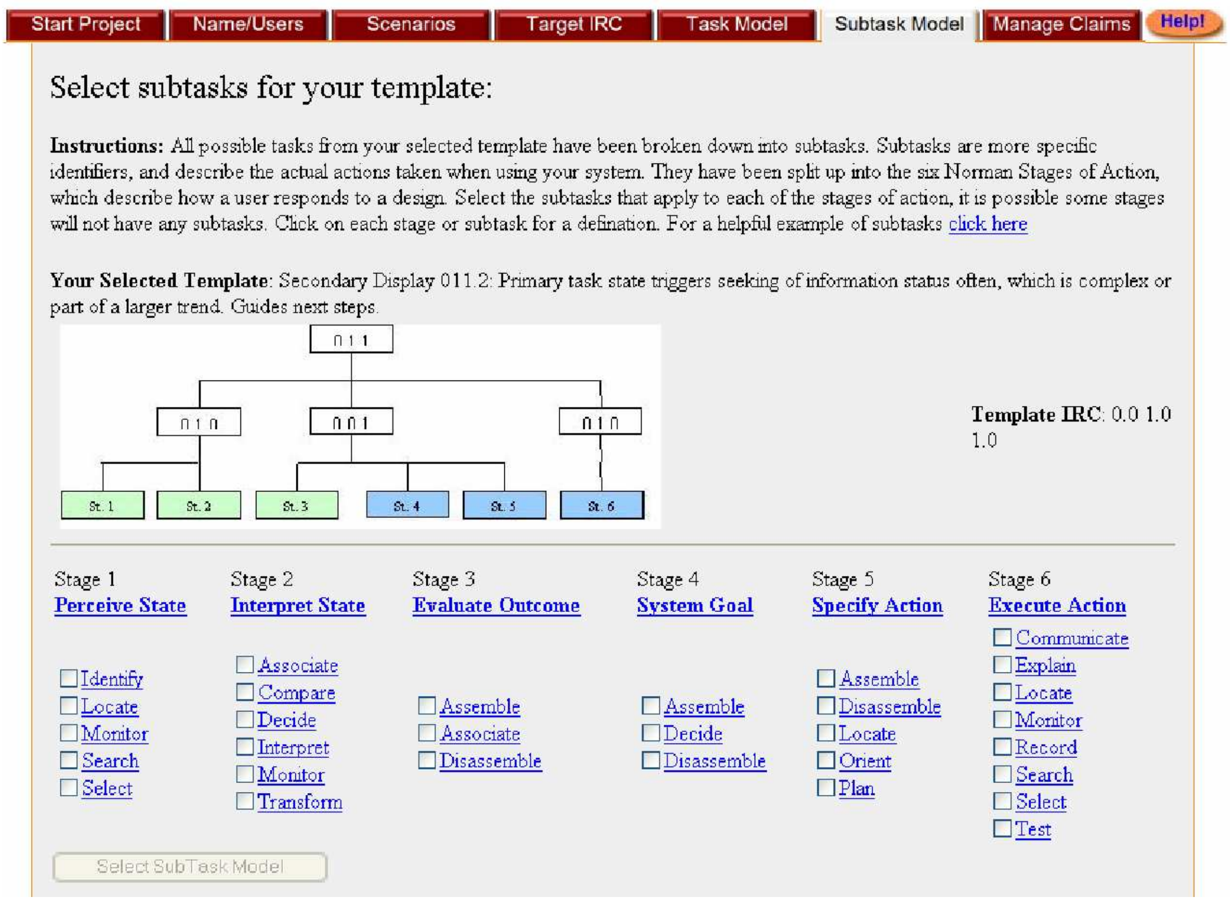


Figure C.8: In the step shown here, the designer selects the specific generic tasks that occur within each stage of action, as well as a pre-defined sub-task model (not yet implemeted) that indicates likely IRC changes within each stage of action.

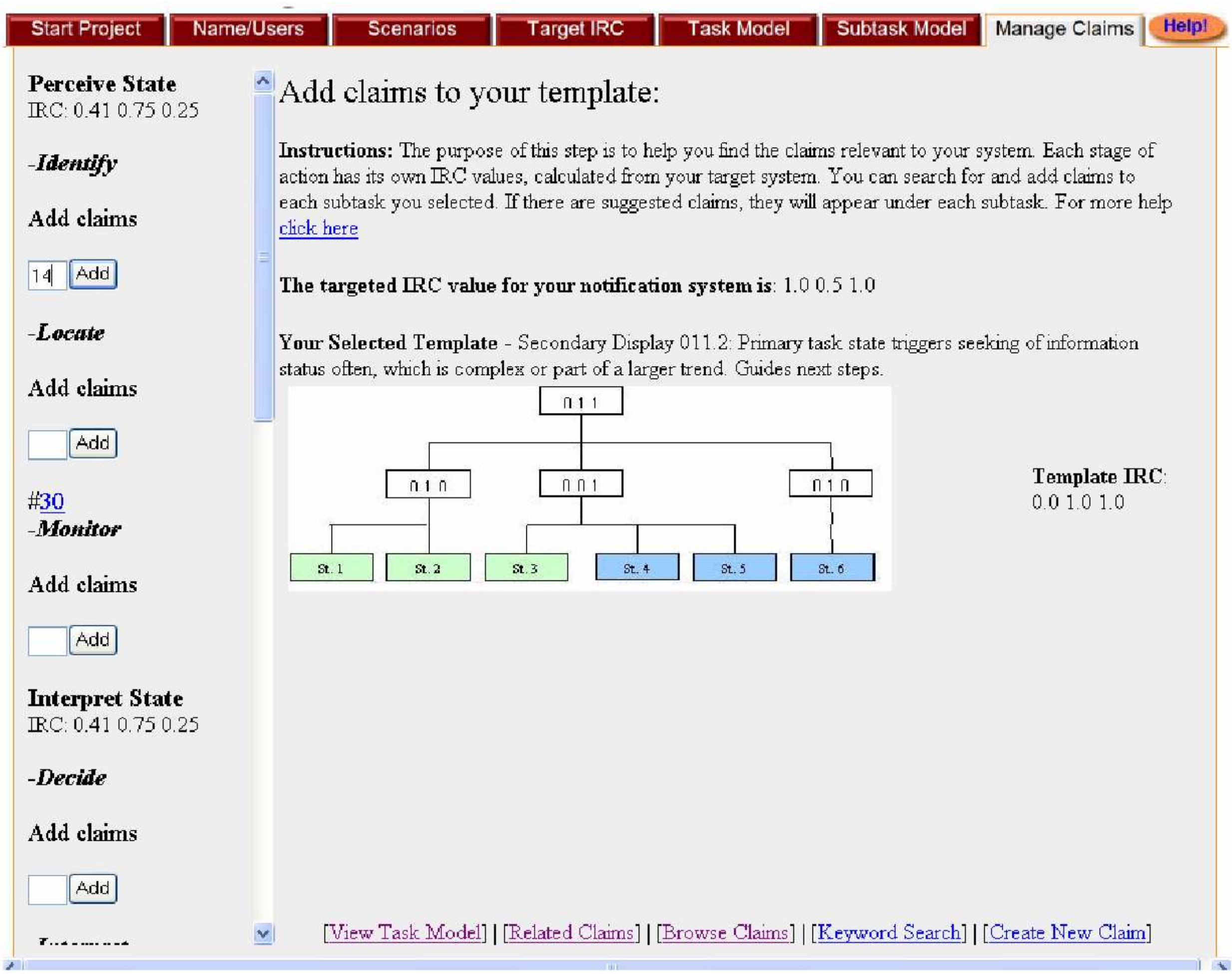


Figure C.9: In this final step, the designer selects or creates new problem claims that elaborate on the sub-tasks within each stage of action (defined in the previous step). Claim possibilities for each sub-task can be recommended, since the IRC characteristics have been defined, as well as the generic tasks. This process results in a problems claim list according to stage of action.

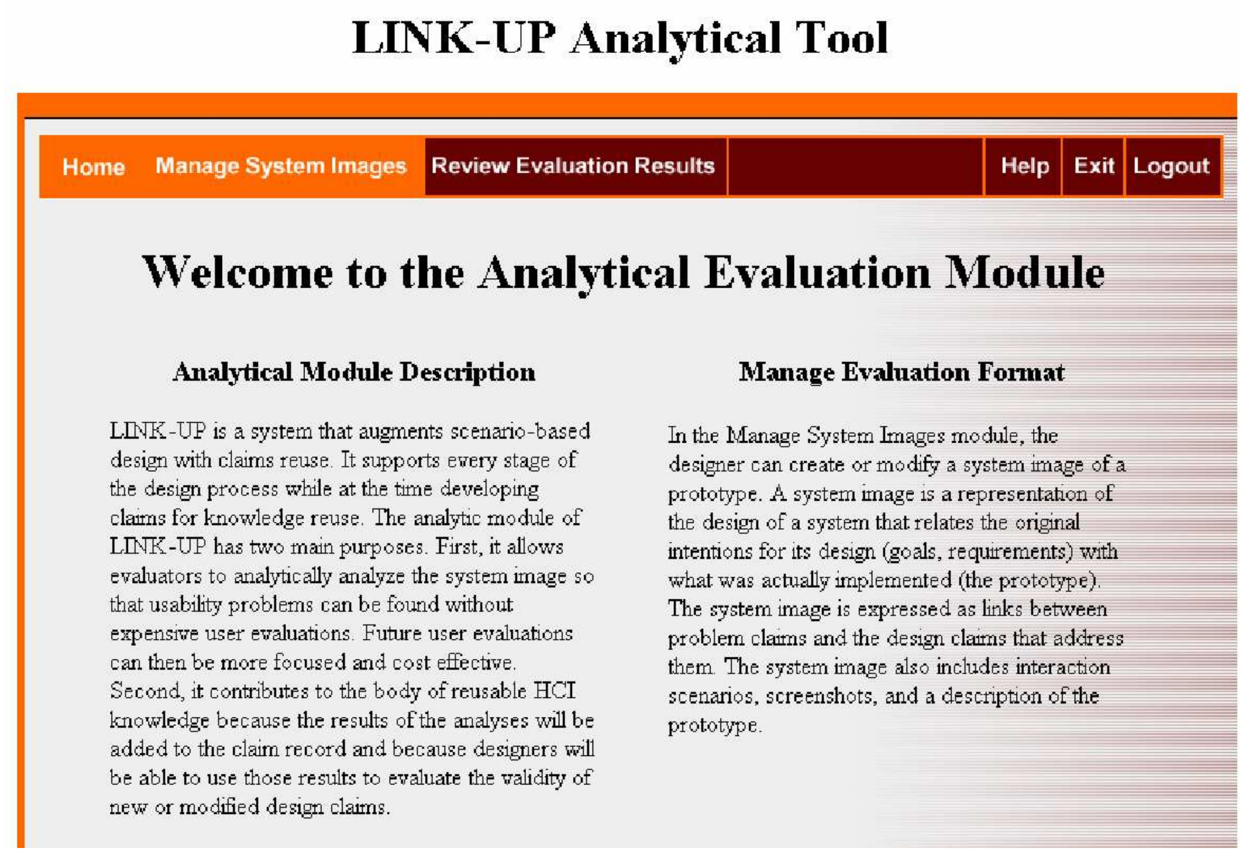


Figure C.10: As one of the initial states within the Analytic Module, this begins the process of the designer creating a record of their system image. From here, the designer can later access the evaluation results, once they have been completed by the expert evaluator.

C.2.2 Analytic Module screenshots

The following screens were developed and used for pilot testing by a team consisting of Jason Chong Lee, Sirong Lin, Alan Fabian, and Andrew Jackson. The team developed the module in JSP with Java servlets; technical assistance and server administration was provided by Edwin Bachetti. As with other design-process modules of LINK-UP, the project was specified and advised by Christa Chewar and Scott McCrickard.

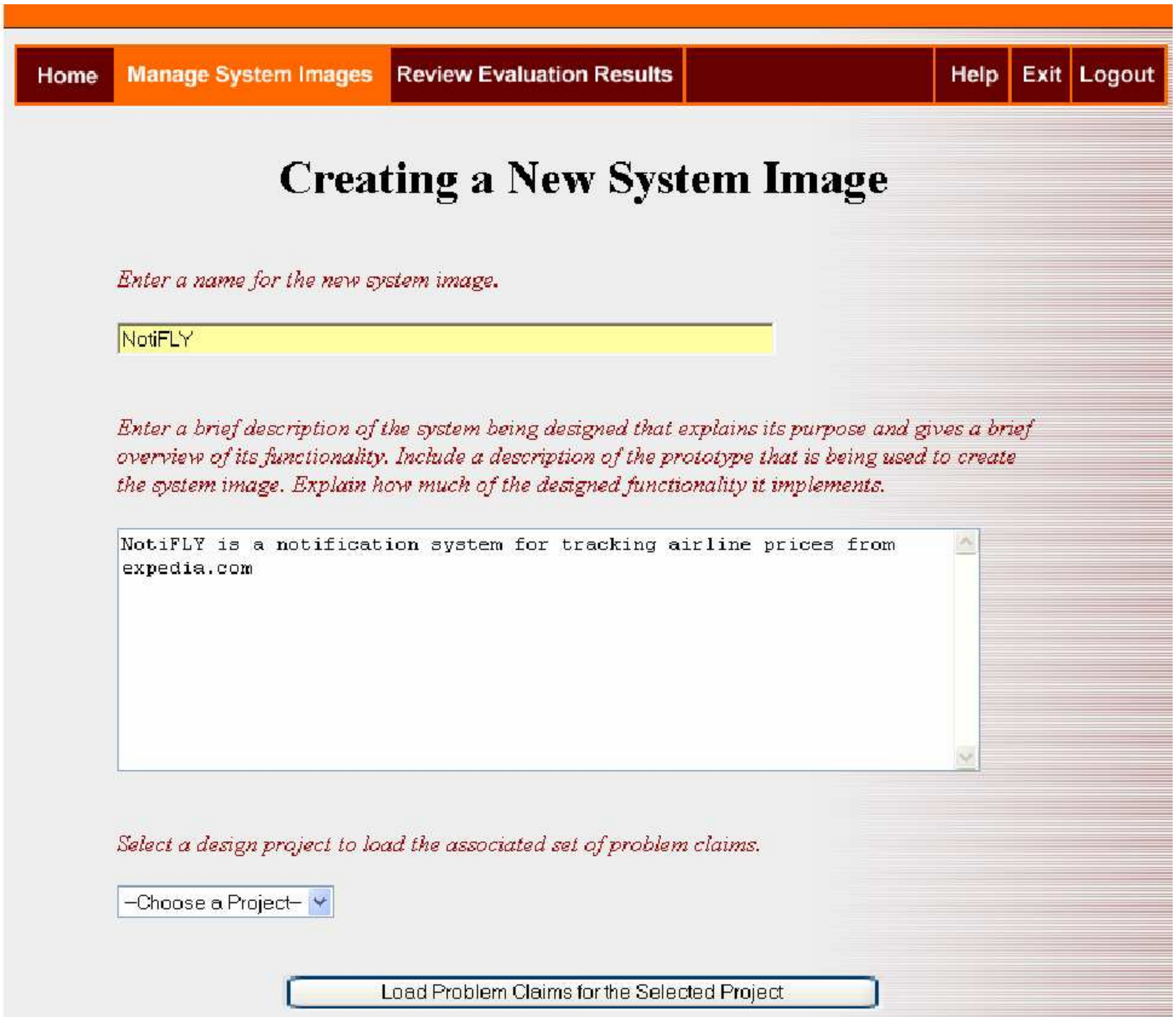


Figure C.11: The designer begins the system image creation process by entering administrative data about their design and associating the to-be-created system image with an existing project file.

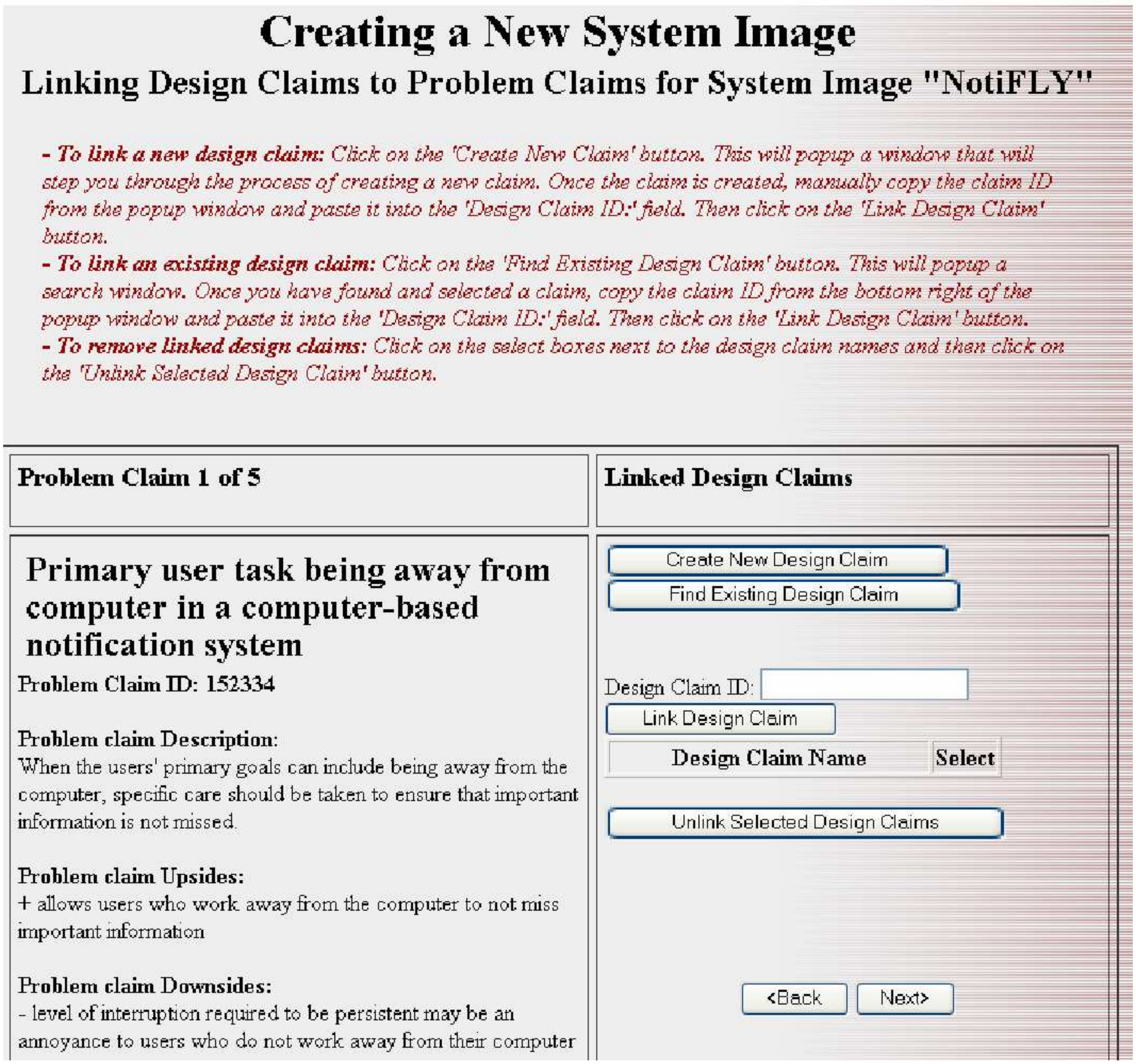


Figure C.12: After associating the system image with an existing problem file, the problem claim set (established in the Requirements Analysis Module) is loaded. Here, the designer enters new or selects existing design claims that address each problem claim. This process is continued for each problem claim.

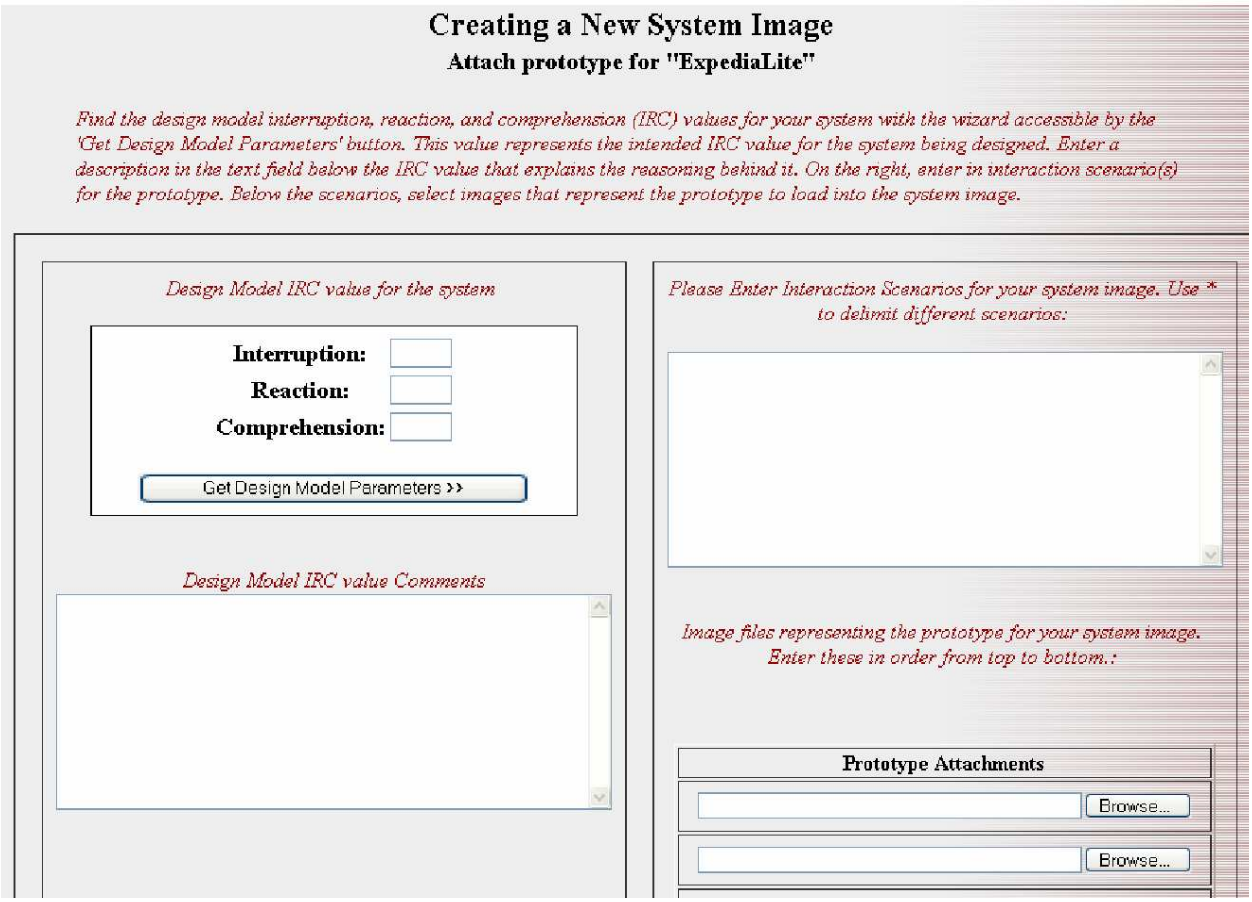


Figure C.13: Once all of the problem claims and design claims have been entered into the system and linked, the designer is able to modify or validate their design model IRC (using IRCspec), enter interaction scenarios describe how their prototype works (or is intended to work), and upload descriptive screenshots, movies, or other material.

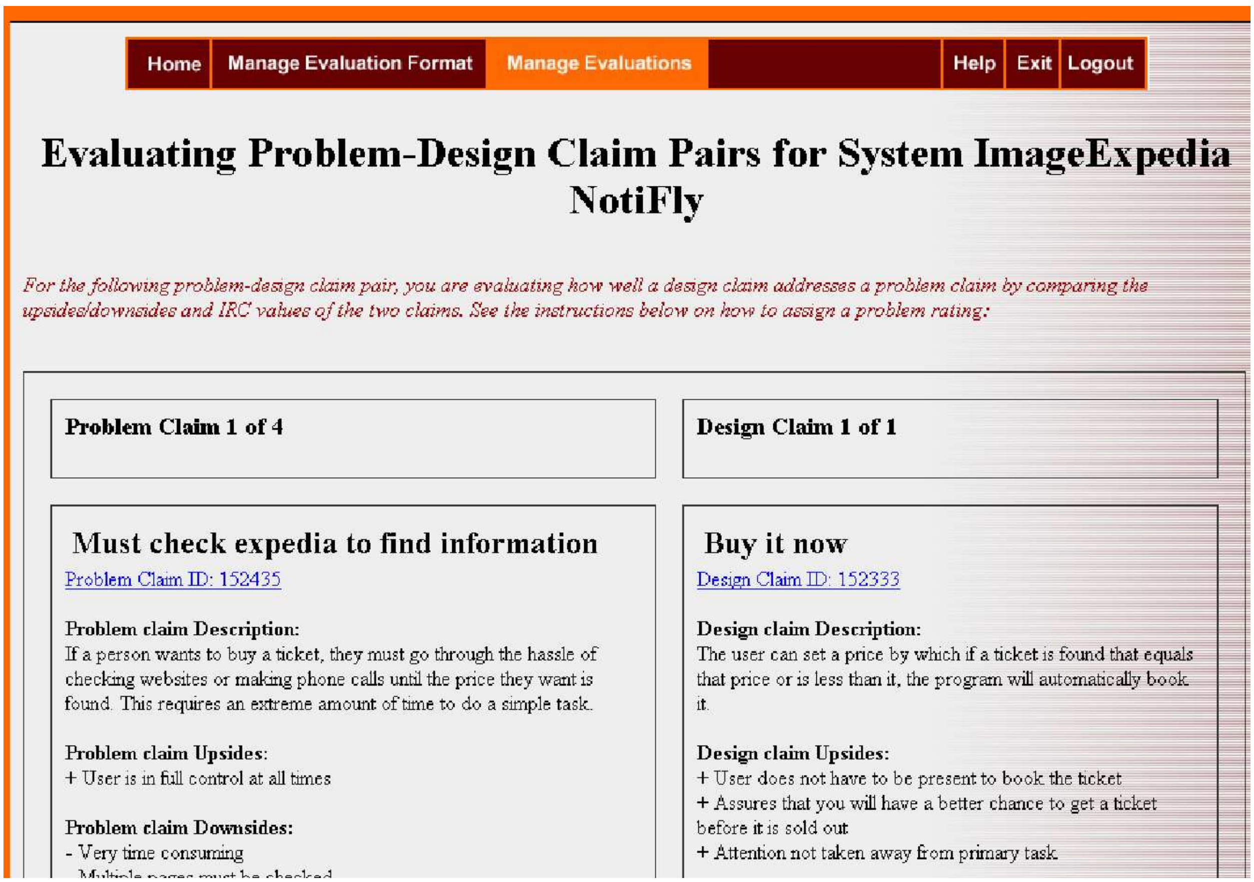


Figure C.14: *With the system image established and made ready for evaluation, an expert evaluator is now able to enter the system, access the system image, and inspect design rationale and descriptive material. Here, the evaluator is able to step through the problem-design claim pairs.*

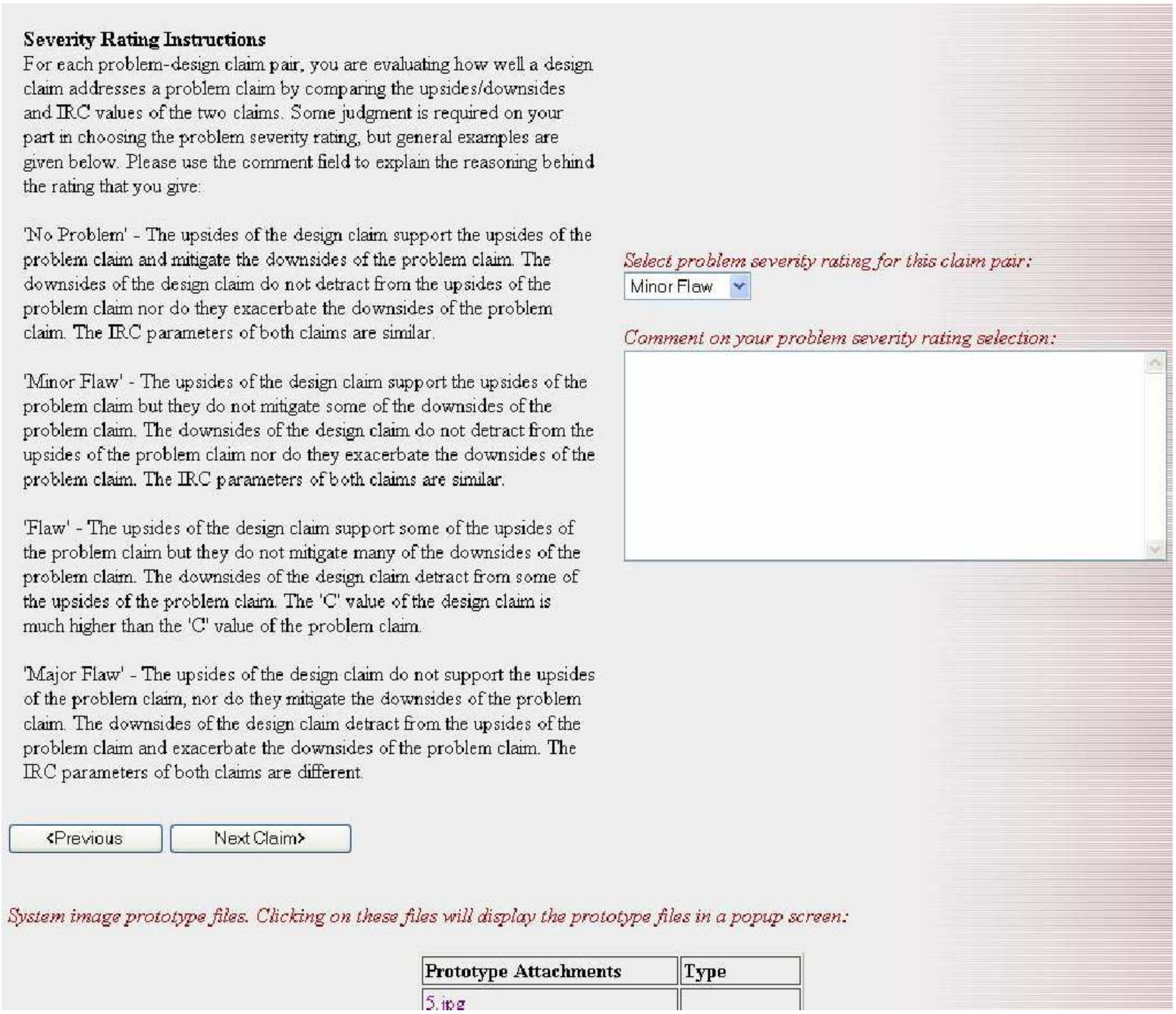


Figure C.15: For each problem-design claim pair in the claim set, the evaluator can indicate whether the claims represent a design or usability flaw (commenting as appropriate) and inspect the associated attachments.

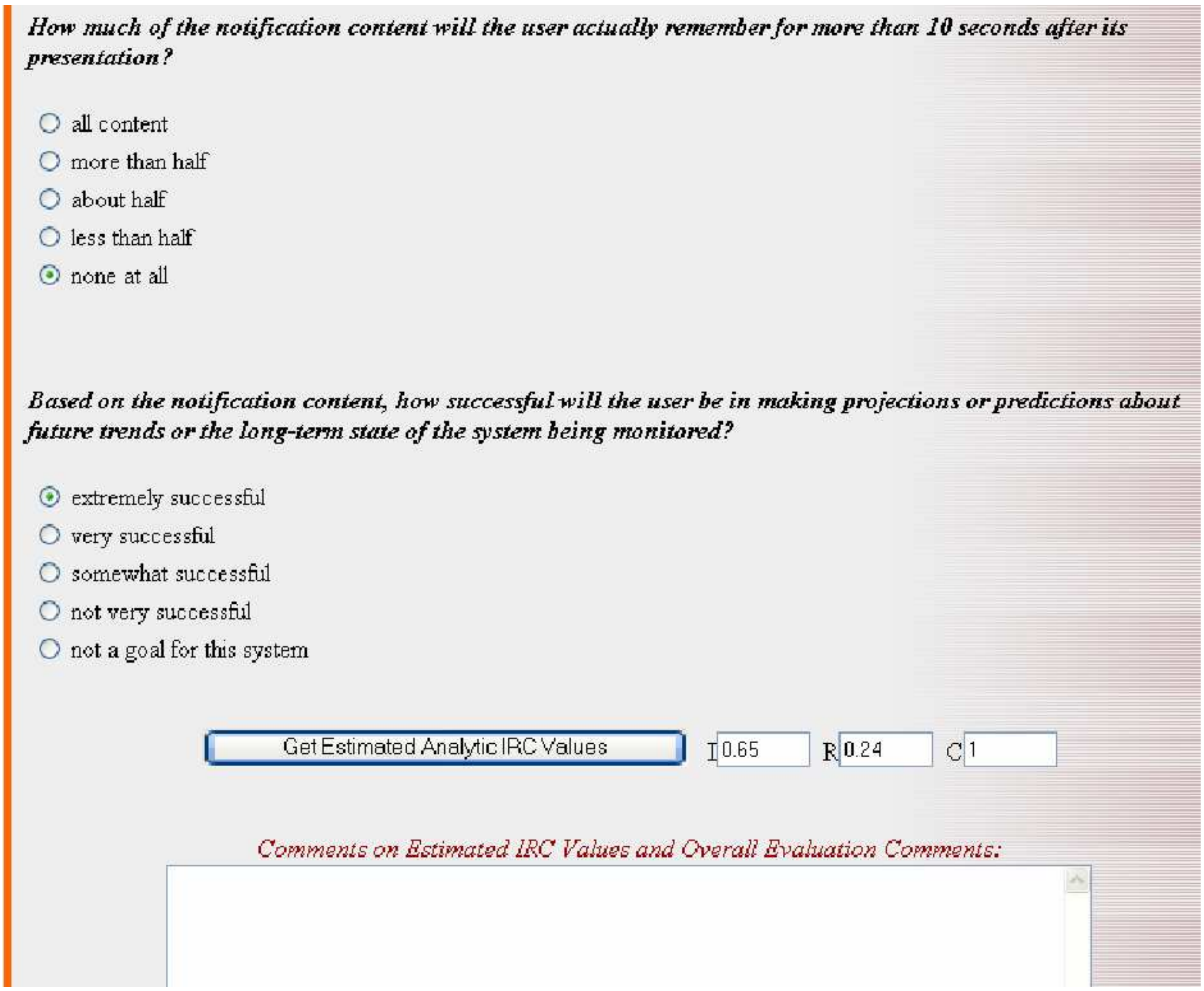
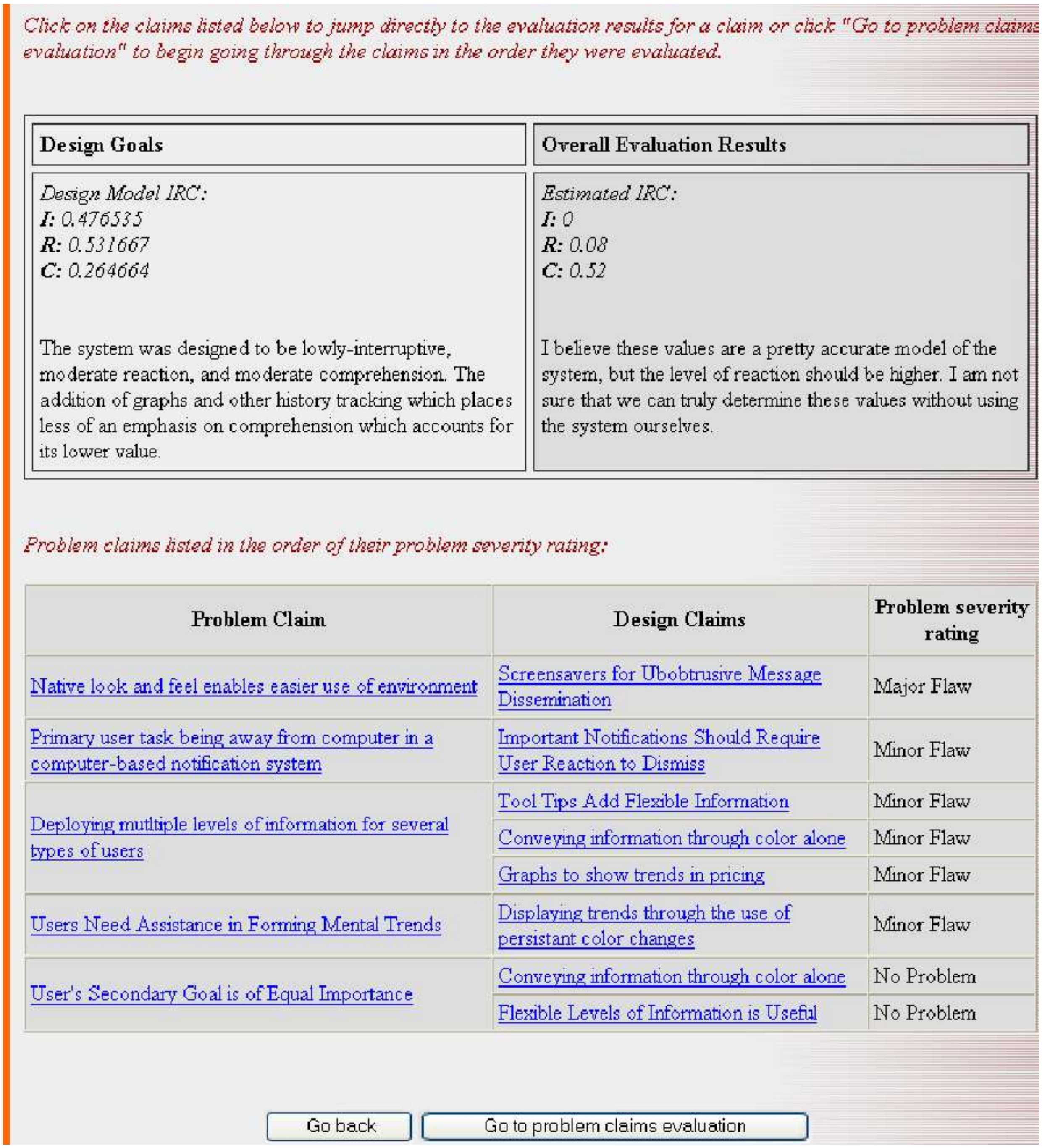


Figure C.16: After inspecting the full claim set, interaction scenarios, and supporting materials (screenshots, etc), the evaluator completes the analytic IRCresults questions to obtain an analytic model IRC. These values and associated comments to clarify the rating become part of the evaluation record.



Click on the claims listed below to jump directly to the evaluation results for a claim or click "Go to problem claims evaluation" to begin going through the claims in the order they were evaluated.

Design Goals	Overall Evaluation Results
Design Model IRC: ***I:*** *0.476535* ***R:*** *0.531667* ***C:*** *0.264664* The system was designed to be lowly-interruptive, moderate reaction, and moderate comprehension. The addition of graphs and other history tracking which places less of an emphasis on comprehension which accounts for its lower value.	*Estimated IRC:* ***I:*** *0* ***R:*** *0.08* ***C:*** *0.52* I believe these values are a pretty accurate model of the system, but the level of reaction should be higher. I am not sure that we can truly determine these values without using the system ourselves.

Problem claims listed in the order of their problem severity rating:

Problem Claim	Design Claims	Problem severity rating
Native look and feel enables easier use of environment	Screensavers for Ubobtrusive Message Dissemination	Major Flaw
Primary user task being away from computer in a computer-based notification system	Important Notifications Should Require User Reaction to Dismiss	Minor Flaw
Deploying mutltiple levels of information for several types of users	Tool Tips Add Flexible Information	Minor Flaw
	Conveying information through color alone	Minor Flaw
	Graphs to show trends in pricing	Minor Flaw
Users Need Assistance in Forming Mental Trends	Displaying trends through the use of persistant color changes	Minor Flaw
User's Secondary Goal is of Equal Importance	Conveying information through color alone	No Problem
	Flexible Levels of Information is Useful	No Problem

Go back | Go to problem claims evaluation

Figure C.17: Here, the designer is provided with a report of the evaluator findings. These findings include a comparison of the design model and analytic model ("Estimated") IRCs, as well as a summary of the flaws found with the claim set.

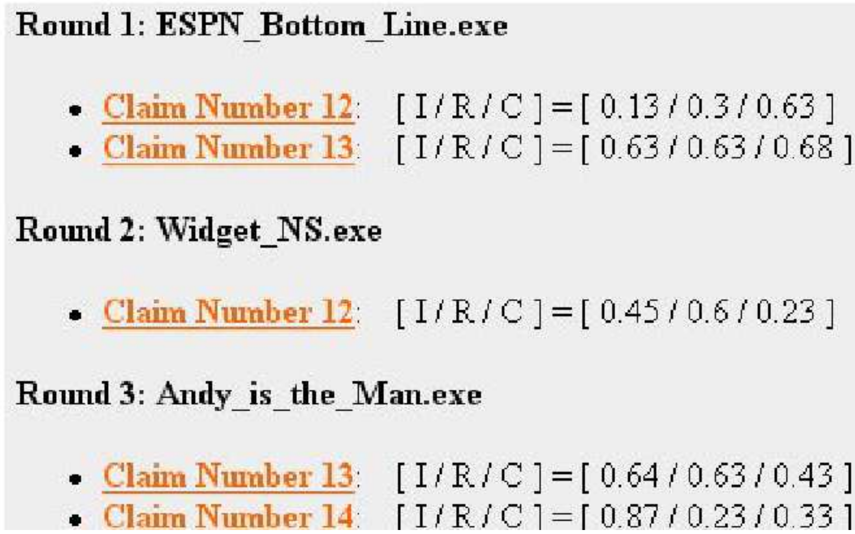


Figure C.18: After executing the empirical testing and uploading the participant's output files, the Empirical Testing Module parses the files and separates IRC data according to each claim, as defined by the designer's original specification.

C.2.3 Empirical Testing Module screenshots

The images depict key portions of the screens were developed and used for pilot testing by a team consisting of Anderson Ray Tarpley III, John Booker, Laurian Hobby, and Jason Zietz. The team developed the module in JSP with Java servlets; technical assistance and server administration was provided by Edwin Bachetti. As with other design-process modules of LINK-UP, the project was specified and advised by Christa Chewar and Scott McCrickard.

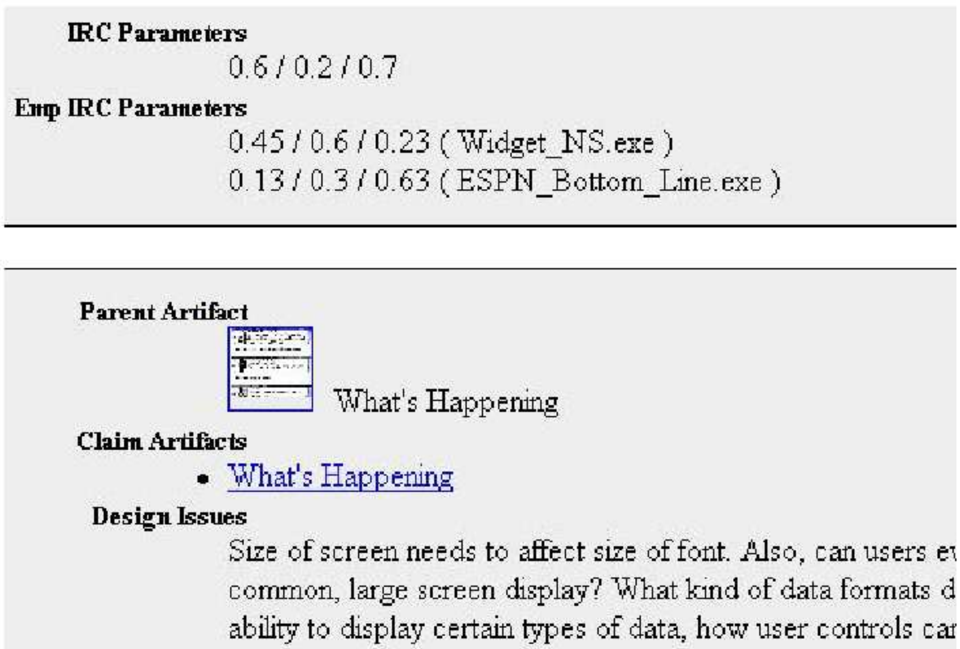


Figure C.19: After the empirical testing results are viewed by the designer, they can be permanently saved to the associated claim records. The system appends the empirically determined IRCs to the claim record in a way that preserves the testing context (i.e., the "Widget_NS.exe" would be a link to the empirical test definition, reusable test script, and test platform.)

Testing my prototype with users provided important findings, either showing the interface was already well-designed or helping to identify weaknesses.

Running a user test with a pre-built platform (like the one we used) was a waste of my time.

I would be very interested to compare my IRC-related results to those obtained by other seminar students with their own prototypes.

If the IRCs obtained by users in this test did not match the IRC I designed the system to achieve, I think it would be very important to redesign my interface or rethink my requirement assumptions.

Having IRC values that characterize user performance <u>does not</u> give me any useful information, if I were to continue designing this interface or others like it.

Basing the initial interface design on a sample set of data and testing a rapid prototype with a standard, pre-built testing platform is a valuable and efficient method of testing and comparing notification artifacts.

I think there would be more benefits to using a testing approach like this if the supporting materials were improved (i.e. the sample Expedia data, the script creator, the testing platform, and the displays of results).

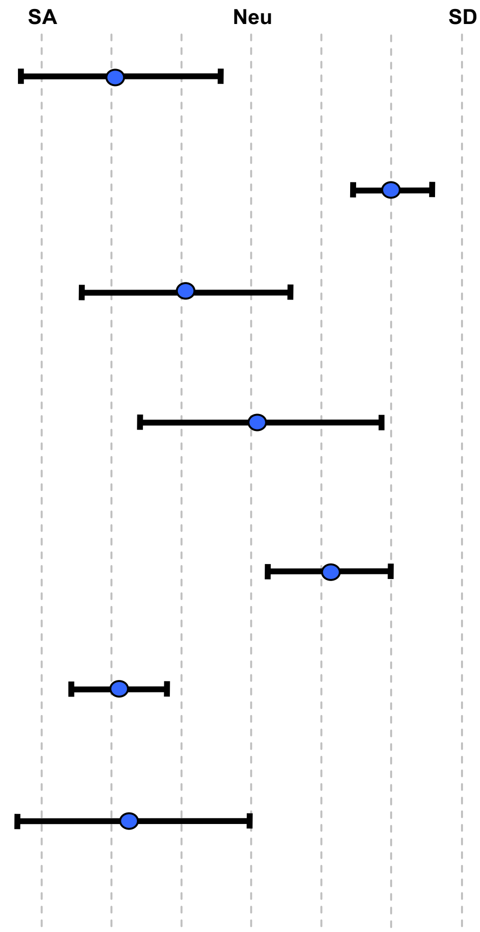


Figure C.20: *Responses related to the Empirical Testing Module from the seven designers that participated in the pilot testing. Means and standard deviations are shown on a scale from "strongly agree" (SA) to "strongly disagree" (SD).*

C.2.4 Selected pilot testing results

Compare the results obtained through the analytical/expert evaluation to the results obtained through the empirical user study:

Which method…. ***(indicate the best choice)***

	Analytical	Empirical	No Difference
Was more difficult to set-up?	☑☑	☑☑☑☑	☑
Was more time-efficient overall?	☑☑☑	☑☑☑☑	
Provided most trustworthy results?	☑☑	☑☑☑☑	☑
Tested important usability aspects?	☑☑	☑☑☑	☑☑
Revealed more usability problems?	☑☑☑	☑☑	☑☑
Provides the best measure of IRC?	☑☑☑☑	☑☑☑	

	Analytical	Empirical	Both	Neither
Would you select to use again?	☑		☑☑☑☑☑☑	
Would you select to get additional result sets?		☑☑☑☑	☑☑☑	

Figure C.21: Responses from the seven designers that participated in the pilot testing for both the Analytic Module and the Empirical Testing Module.

Christa M. Chewar

Academic Background:

Virginia Polytechnic Institute and State University, Blacksburg, VA

Ph.D.	Computer Science	2005 (ABD)	GPA: 3.93
M.S.	Computer Science	2003	GPA: 3.92

Fellowship awarded by the U. S. Army for Advanced Civil Schooling, Fall 2001 – Spring 2004

Webster University, St. Louis, MO

M.B.A	Business Administration	2001	GPA: 3.95

United States Military Academy, West Point, NY

B.S.	Computer Science (FOS, Info Systems) Geography (dual-major)	1995	GPA: 3.36, In major: 3.95

Distinguished Cadet 1994, 1995
Veterans of Foreign Wars Award for Excellence in the Computer Science Sequence

Military Schools:

Combined Arms Staff Service School	1999	Fort Leavenworth, KS
Signal Officer Captain's Career Course	1999	Fort Gordon, GA
Signal Officer Basic Course	1995	Fort Gordon, GA

Positions Held:

Officer, U.S Army (Active-duty) 1995 – current

Instructor, Computer Science	2004 – current	West Point, NY
Company Commander, U.S. Army	1999 – 2001	Fort Jackson, SC
Telecommunications Plans Officer, U.S. Army	1998	Mannheim, Germany
Battalion Logistics Officer, U. S. Army	1997	Mannheim, Germany
Platoon Leader, U.S. Army	1996	Kaposvar, Hungary

Teaching Experience

Dept of Electrical Engineering & Computer Science, 2004 – current
U. S. Military Academy, West Point, NY

Refereed Conference and Journal Publications:

C. M. Chewar, D. Scott McCrickard, and John M. Carroll. “Analyzing the Social Capital Value Chain in Community Networks.” *Journal of Internet Research*. To appear.

D. Scott McCrickard and C. M. Chewar. “Designing Attention-Centric Notification Systems: Five HCI Challenges.” *Cognitive Systems: Human Cognitive Models in Systems Design.* Ed. J. Chris Forsythe. Lawrence Earlbaum, 2005. To appear.

Cyril Montabert, Dillon Bussert, Soloman Gifford, C. M. Chewar, and D. Scott McCrickard. “Supporting Requirements Reuse in Notification Systems Design Through Task Modeling.” To appear in *Proceedings of the 11th International Conference on Human-Computer Interaction (HCII '05)*, Las Vegas NV, July 2005.

Ali Ndiwalana, Jason Lee, Jamie L. Smith, Shahtab Wahid, Laurian Hobby, C. M. Chewar, and D. Scott McCrickard. “From Personas to Design: Creating a Collaborative Multi-disciplinary Design Environment.” To appear in *Proceedings of the 11th International Conference on Human-Computer Interaction (HCII '05)*, Las Vegas NV, July 2005.

Jason Chong Lee, C. M. Chewar, and D. Scott McCrickard. “Image is Everything: Advancing HCI Knowledge and Interface Design Using the System Image.” In *Proceedings of the ACM Southeast Conference (ACMSE '05)*, Kennesaw GA, March 2005, pp. 2-376 - 2-381.

Laurian Hobby, John E. Booker, D. Scott McCrickard, C. M. Chewar, and Jason Zietz. “Facilitating and Automating Empirical Evaluation for Education.” In *Proceedings of the ACM Southeast Conference (ACMSE '05)*, Kennesaw GA, March 2005, pp. 1-303 - 1-308.

C. M. Chewar and D. Scott McCrickard. “Links for a Human-Centered Science of Design: Integrated Design Knowledge Environments for a Software Development Process.” In *Proceedings of the 2005 Hawai'i International Conference on System Sciences (HICSS '05)*, Waikoloa HI, January 2005, 10 pgs (CD-ROM).

Shahtab Wahid, Jamie L. Smith, Brandon Berry, C. M. Chewar, and D. Scott McCrickard. “Visualization of Design Knowledge Component Relationships to Facilitate Reuse.” In *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration (IRI '04)*, Las Vegas NV, November 2004, pp. 414-419.

Jason Chong Lee, Sirong Lin, C. M. Chewar, and D. Scott McCrickard. “From Chaos to Cooperation: Teaching Analytic Evaluation with LINK-UP.” In *Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (E-Learn '04)*, Washington DC, November 2004, pp. 2755-2762.

Jacob P. Somervell, C. M. Chewar, and D. Scott McCrickard. “Making a Case for HCI: Comparing Materials for Case-Based Teaching.” *Proceedings of the 2004 Conference on Frontiers in Education (FIE ‘04)*, Savannah, GA, October 20-23, 2004, 5 pages.

C. M. Chewar, D. Scott McCrickard, and Alistair G. Sutcliffe. “Unpacking Critical Parameters for Interface Design: Evaluating Notification Systems with the IRC Framework.” *Proceedings of the 2004 Conference on Designing Interactive Systems (DIS '04)*, Cambridge MA, August 1-4, 2004, 10 pages.

Shahtab Wahid, C. M. Allgood, C. M. Chewar, and D. Scott McCrickard. “Entering the Heart of Design—Relationships for Tracing Claim Evolution.” *Proceedings of the 2004 Conference on Software Engineering and Knowledge Engineering (SEKE '04)*, Banff, Alberta, Canada, June 21-24, 2004, 6 pages.

John E. Booker, C. M. Chewar, and D. Scott McCrickard. "Usability Testing of Notification Interfaces: Are We Focused on the Best Metrics?" *Proceedings of the ACM Southeast Conference (ACMSE '04)*, Huntsville AL, April 2004, pp. 128-133.

Alan Fabian, David Felton, Melissa Grant, Cyril Montabert, Kevin Pious, Nima Rashidi, Anderson Ray Tarpley III, Nicholas Taylor, C. M. Chewar, and D. Scott McCrickard. "Designing the Claims Reuse Library: Validating Classification Methods for Notification Systems." *Proceedings of the ACM Southeast Conference (ACMSE '04)*, Huntsville AL, April 2004, 357-362.

D. Scott McCrickard and C. M. Chewar. "Proselytizing Pervasive Computing Education: A Strategy and Approach Influenced by Human-Computer Interaction." In *Proceedings of the Second IEEE Conference on Pervasive Computing and Communications Workshops (PerEd '04)*, Orlando FL, March 2004, pp 257-262.

D. Scott McCrickard, C. M. Chewar, and Jacob P. Somervell. "Design, Science, and Engineering Topics--Teaching HCI with a Unified Method." In *Proceedings of the 2004 ACM Technical Symposium on Computer Science Education (SIGCSE '04)*, Norfolk VA, March 2004, pp 31-35.

C. M. Chewar, Edwin Bachetti, D. Scott McCrickard, and John Booker. "Automating a Design Reuse Facility with Critical Parameters: Lessons Learned in Developing the LINK-UP System." In *Proceedings of the 2004 International Conference on Computer-Aided Design of User Interfaces (CADUI '04)*, Island of Madeira, Portugal, January 2004, pp. 236-247.

D. Scott McCrickard, C. M. Chewar, Jacob P. Somervell, and Ali Ndiwalana. "A Model for Notification Systems Evaluation—Assessing User Goals for Multitasking Activity." *ACM Transactions on Computer-Human Interaction (TOCHI) 10* (4), December 2003, pp. 312-338.

Catherine Payne, C. F. Allgood, C. M. Chewar, Chuck Holbrook, and D. Scott McCrickard. "Generalizing Interface Design Knowledge: Lessons Learned from Developing a Claims Library." In *Proceedings of the 2003 IEEE International Conference on Information Reuse and Integration (IRI '03)*, Las Vegas NV, October 2003, pp. 362-369.

C. M. Chewar and D. Scott McCrickard. "Educating Novice Developers of Notification Systems: Targeting User-Goals with a Conceptual Framework." In *Proceedings of the World Conference on Educational Multimedia/Hypermedia and Educational Telecommunications (ED-MEDIA '03)*, Honolulu HI, June 2003, pp. 2759-2766.

Douglas G. Turnbull, C. M. Chewar, and D. Scott McCrickard. "Are Cognitive Architectures Mature Enough to Evaluate Notification Systems?" In *Proceedings of the 2003 International Conference on Software Engineering Research and Practice (SERP '03)*, Las Vegas NV, June 2003 (7 pages, CD-ROM).

D. Scott McCrickard, Richard Catrambone, C. M. Chewar, and John T. Stasko. "Establishing Tradeoffs that Leverage Attention for Utility: Empirically Evaluating Information Display in Notification Systems" *International Journal of Human-Computer Studies 8* (5), May 2003, pp. 547-582.

D. Scott McCrickard and C. M. Chewar. "Attuning Notification Design to User Goals and Attention Costs." *Communications of the ACM* , 46 (3), March 2003, pp. 67-72.

Ali Ndiwalana, C. M. Chewar, Dillon Bussert, Jacob Somervell, and D. Scott McCrickard. "Ubiquitous Computing: By the People, For the People." In *Proceedings of the ACM Southeast Conference (ACMSE 2003)*, Savannah GA, March 2003, pp. 16-21.

Jacob Somervell, C. M. Chewar, D. Scott McCrickard, and Ali Ndiwalana. "Enlarging Usability for Ubiquitous Displays." In *Proceedings of the ACM Southeast Conference (ACMSE 2003)*, Savannah GA, March 2003, pp. 24-29.

C. M. Chewar, and D. Scott McCrickard. "Dynamic Route Descriptions: Tradeoffs by Usage Goals and User Characteristics" In *Proceedings of the International Symposium on Smart Graphics (Smart Graphics 2002)*, Hawthorne, NY, June 2002, pp. 71-78.

C. M. Chewar, D. Scott McCrickard, Ali Ndiwalana, Chris North, Jon Pryor, and David Tessendorf. "Secondary Task Display Attributes -- Optimizing Visualizations for Cognitive Task Suitability and Interface Avoidance." In *Proceedings of the IEEE TCVG Symposium of Visualization (VisSym 2002)*,Barcelona Spain, May 2002, pp. 165-171.

Jacob Somervell, C. M. Chewar, and D. Scott McCrickard. "Evaluating Graphical vs Textual Secondary Displays for Information Notification." In *Proceedings of the ACM Southeast Conference (ACMSE 2002)*, Raleigh NC, April 2002, pp. 153-160.

Refereed Short Papers:

Ali Ndiwalana, C. M. Chewar, Jacob Somervell, and D. Scott McCrickard. "Ubiquitous Computing: By the People, For the People." Short paper in *Proceedings of the 2003 ACM Conference on Human Factors in Computing Systems (CHI 2003)*, Fort Lauderdale FL, April 2003, pp. 968-969.

C. M. Chewar and D. Scott McCrickard. "Evaluating Information Design for Notification Systems." Extended abstract in *Proceedings of the 24th Annual Meeting of the Cognitive Science Society (CogSci2002)*, Fairfax, VA, August 8-10, 2002, pp. 995.

David Tessendorf, C. M. Chewar, Ali Ndiwalana, Jon Pryor, D. Scott McCrickard, and Chris North. "An Ordering of Secondary Task Display Attributes." Interactive poster in *Conference Companion of the ACM Conference on Human Factors in Computing Systems (CHI 2002)*, Minneapolis MN, April 2002, pp. 600-601.

Technical Reports:

Jacob Somervell, C.M. Chewar, D. Scott McCrickard. "Making a Case for HCI: Comparing Materials for Case-Based Teaching." Technical Report TR-04-07, Computer Science, Virginia Tech.

C. M. Chewar and D. Scott McCrickard. "Claims Reuse for Notification Systems Design: LINK-UP Vision and IRC Equations." Technical Report TR-03-29, Computer Science, Virginia Tech.

C. M. Chewar, D. Scott McCrickard, and John M. Carroll. "Persistent Virtual Identity in Community Networks: Impact to Social Capital Value Chains." Technical Report 03-01, Computer Science, Virginia Tech.