

NotiFly: Enhancing Design through Claims-Based Personas and Knowledge Reuse

Justin Belcher, Raheel Aidrus, Ben Congleton, Doug Hall, Shahzad Hussain, Matthew Jablonski, Theresa Klunk, and D. Scott McCrickard

Center for Human Computer Interaction and Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, VA USA 24061-0106

{jubelche, raidrus, bconglet, dohall, shussain, mjablons, tklunk, mccricks} @ vt.edu

ABSTRACT

Typically viewed as competing design approaches, this paper illustrates how claims and personas can be used together in user interface design. Our combined approach is exemplified in the development of a notification system called NotiFly. We introduce the idea of claims-based personas as a design tool in a scenario-based approach. Following a discussion of NotiFly's development, we speculate on the theories that led to the system development and how they might impact HCI design process. Based on the findings encountered during the development of NotiFly, we present direction for future work to better systems like Notifly and notification systems in general.

Categories and Subject Descriptors

H.1.1 [Models and Principles]: Systems and Information Theory: *General systems theory*; H.1.2 [Models and Principles]: Systems and Information Theory: *Human Factors*; D.2.2 [Software Engineering]: Design Tools and Techniques: *Evolutionary prototyping*

General Terms

Performance, Design, Experimentation, Human Factors

Keywords

Notification system, claim, persona, reuse, scenario, interface design, human computer interaction

1. INTRODUCTION

The discipline of human computer interaction (HCI) is fast becoming an important research focus in software and usability engineering. HCI studies range from how the user responds to interface choices such as color and layout, to interaction

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

43rd ACM Southeast Conference, March 18-20, 2005, Kennesaw, GA, USA. Copyright 2005 ACM 1-58113-000-0/00/0004...\$5.00.

techniques, to the selection of critical task-related activities [8]. This focus is particularly important in the emerging field of notification systems—where important information can be misinterpreted or lost if not presented effectively.

At the heart of HCI lies the theories of user-centrality—theories that are constantly evolving as the discipline exits its infancy. It is often that when those theories are analyzed and exercised that new insight can be drawn from them. In this paper, we will discuss several such insights that occurred during the development of a notification system called NotiFly.

NotiFly is a software system built to monitor online flight pricesⁱ. The purpose behind developing NotiFly was more deeply-rooted in evaluating and expanding scenario-based design than creating a novel application, but the latter occurred as a result of the design processes it followed.

NotiFly was developed over the course of a semester by seven undergraduate computer science students with an interest in human computer interaction. From the design and development process that took place, several key issues in HCI were visited:

1. How can differing design models developed independently be unified and streamlined through claims and reuse?
2. What improvements can be made upon scenario-based designs that specifically pertain to notification systems?
3. In what ways can attentive notification systems be implemented and enhanced through other tools of design?

In this discussion, we will reflect on the development of NotiFly. It is our hope that by revisiting this process, designers and instructors will gain insight on how to structure development that is conducive to exploring current HCI theories.

2. BACKGROUND

This study encompasses several key approaches and concepts taken directly from human computer interaction. In this section,

ⁱ The system specifically monitored <http://www.expedia.com>

we define and explain these principles to provide the reader with necessary context for discussion.

2.1 What are Personas?

An important design tool utilized during NotiFly's development was a user persona. A *persona* is a user archetype which guides decisions about product features, navigation, interactions, and even visual design [3]. Generally, personas describe a real potential user of the system that is being developed. Doing so allows designers to immediately consider not only the users' goals and needs, but their behavior and expectations as well. Designing for these archetypes also simultaneously satisfies the requirements of a broader group of users who share the same characteristics [4]. An example of a persona used in developing a word processor could be an elderly man named Clyde. Clyde is generally apprehensive about technology and dislikes feeling incompetent as a result. Clyde's main purpose for using a word processor is to decrease the amount of time it takes to draft documents by hand, but does not want to be faced with a steep learning curve to achieve those gains in productivity. Structuring your design model with Clyde in mind might deter you from cluttering the interface with unneeded functionality or prompt you to place more importance in user-friendly help systems.

2.2 What is Scenario-Based Design?

Scenario-based design is a usability engineering paradigm in which human characteristics and needs play a pivotal role in the design process [10]. Scenarios tell the story of a potential user interacting with your system, and scenarios typically include actors, goals, and sequences of processes in which goals are attained, modified, or abandoned [1, 10]. Under this model, user scenarios surpass simple examples and background data to become workable objects of design. These scenarios are constantly revisited and refined through an iterative cycle, and are later verified by empirical evaluation [1].

2.2.1 What is a Claim?

A *claim* is a statement that describes the effect a feature will have on a user within a usage scenario [11]. In much the same way artifacts represent physical objects of design—claims represent intellectual objects of design. Claims consist of a title, description, upsides and downsides that relate to the description, scenarios, theories, and artifacts that support the description [1, 10]. These attributes allow designers to compare one claim to another as well as facilitate their reuse through a shared knowledge repository [11]. A potential claim derived from the design of an email client could be the use of a pop-up window for notifications. The designer would describe the claim's use in a small description, list upsides such as quick delivery of information, and cite downsides such as potential user irritation or missed notifications. Not only does this make the designer objectively critique the component's use and ramifications within the system, but it would also allow another designer wanting to implement a similar component to reuse the claim in their design.

3. RESEARCH DOMAIN

The concepts and applications of scenario-based design along with other user-centric models are important issues to HCI.

Several areas of research which have spun off of these topics are relevant to this paper and are discussed below.

3.1 Notification Systems

Notification systems are hardware or software systems designed to deliver relevant information to users engaged in other activities. These systems are a particular concern to HCI because of the current trend of monitoring multiple sources of information concurrently [6]. Some systems that exist within this domain include stock tickers, in-vehicle navigation systems, and real world interfaces such as highway notification displays.

3.2 The IRC Framework

The *IRC Framework* is a system of classification for notification systems in which three critical parameters are considered: interruption, reaction, and comprehension. These parameters represent abstract user goals which are essential to information processing systems such as notification systems [5].

Each of the three critical parameters is expressed quantitatively as a value ranging from 0 to 1. Interruption corresponds to how much user attention is reallocated from a primary task to the secondary system. Reaction involves an instantaneous stimulus response to the interruption. Finally, comprehension describes not only how well the user understands the information presented but also how well that information is retained. An alarm notification system designed to sound an air horn in your bedroom at 5:00AM until you wake up and turn it off would most likely have an IRC value of (1.0, 1.0, 0.25). This value corresponds to the fact that the system requires a complete reallocation of attention from your primary task of sleeping to the alarm system, prompts an almost immediate reaction from the user to disengage the notification, and carries little information to comprehend other than it is time to wake up.

3.3 Implications for Design

The real benefit of the IRC Framework, as it pertains to notification systems, is the ability to express any design model in terms of critical parameters. Claims, scenarios, personas, and other design tools can all contain IRC values. Consider the usefulness of designing a system with claims that support the IRC values of the design model, and then being able to empirically evaluate the effectiveness of the system through user model IRC values gathered during testing. A process such as this greatly aids designers by enabling them the ability to assess the abstract design issues they hope to address.

4. METHODOLOGY

The development of NotiFly was deeply-rooted in claims and scenario-based design. Through the processes of individual prototyping and collaboration, a system arose that was not only innovative in design but also raised some interesting interface design questions left for future work.

4.1 Initial Prototyping

Each student in the undergraduate seminar developed a notification system adhering strictly to scenario-based design for solidifying their design models. It was up to the discretion of the designer to envision specifics about the system including user

goals, target system IRC values, and potential notification system design space [7].

The development of these notification systems included the utilization of reusable claims housed in a claims library as well as artifacts of the students' own invention [8]. Orienting the design of the systems through heavy use of claims enabled the students to assess each aspect of their design model analytically, but the lack of structure defined by the project yielded little consistency from person to person or from individual claim to system. This inconsistency made for somewhat weak design models, but the use of critical parameters as a design tool coupled with extensive discussion of the developer's individual design intents yielded many desirable features.

At this point, the instructor organized two usability evaluations for the group—one as a peer assessment and another as a student-led empirical study. First, each student posted screenshots and design rationale to the same integrated development environment (IDE) [2] that they used to interface with the claims library. This allowed other group members to serve as anonymous expert evaluators for each student's notification system. These evaluations represented a potential user model that could then be quantitatively compared to the original design model [8]. Next, each student conducted an empirical evaluation of their own system in a dual-task situation through a tool in the IDE. This tool monitored participants' activities during both a primary task and notifications received from the students' systems to quantitatively produce an actual user model in terms of critical parameters (see Figure 1).

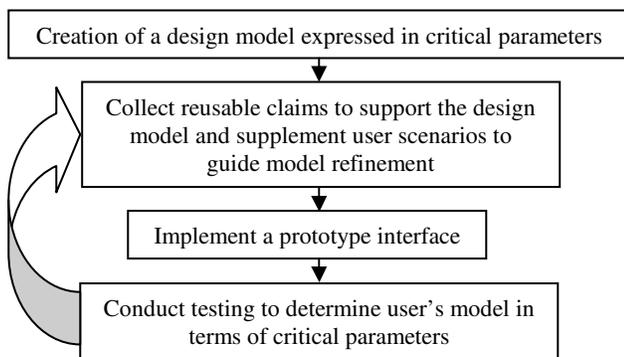


Figure 1. Initial design process of notification systems by adhering to scenario-based design principles [8]

4.2 Group Collaboration

After undergoing this development process, the students in the group were faced with either completing an additional iteration of their system to tighten the gap between their design and user models, or to proceed forward under their own discretion. The latter was agreed upon, and the group decided to make an entirely new notification system through a collaborative effort. This effort would allow the group to reuse claims and design objects collected through their individual prototypes to guide the development of the new system [8].

4.2.1 Design Rationale

The system's design rationale was closely-tied to that of the original student prototypes—a notification system for tracking flight prices. Instead of designing the system to fit closely into one corner of the notification systems design space, the students agreed that variability in the critical parameters of the system through customization would enhance overall usability.

4.2.2 Implementation / Team Delegation

The system was implemented entirely in C#. The reasoning behind this decision largely dealt with the group's familiarity with C++ and the rapid GUI development possible with .NET's Windows Forms library.

Four members of the team were responsible for developing the backend of the application, including the threading structure, the socket and network layer, and the parser required to retrieve flight information from the web. One student was responsible for graphics and user interface development, and the remaining two team members conducted the user and unit testing. A small team coupled with a rapid development cycle would spell disaster for most software engineering efforts, but the direction and structure gained from the design process they followed enabled the students to fulfill all system requirements in a timely fashion.

4.2.3 System Functionality and Rationale

At the onset of the program, the user is tasked with choosing a persona that accurately describes their personality and needs. This allowed NotiFly to custom-cater its interactions based on that persona—a more detailed discussion of this feature is left to a later section. From the client, users enter desired flight information into NotiFly through an interface similar to Expedia.com®. At that point, the system would retrieve the current prices for that flight and begin monitoring changes through the notification system. NotiFly also enabled users to conduct and monitor any number of flight queries that they desired. The notification system itself resided on the side of the user window as a small transparent icon. When new flight information arrived, the icon would expand out to notify the user of the change and provide them with a graph of price trends for all flights that were monitored. When a flight price fell below user-defined threshold, a "buy it now" notification was issued, and the ticket could be purchased through the Expedia.com® web interface.

4.2.4 User Testing

User testing was performed to verify that users of the system were able to select personas that accurately reflected their notification preferences. The NotiFly binary and testing instructions were encapsulated into a single distributable package that was sent to test participants to be installed and tested on their home machines. 26 users were asked to install NotiFly, choose the persona that most closely matched their preferences, configure a few basic flight searches, and resume normal computer usage. Testing users in their natural environment assured that testing results would represent real world usage patterns. After 1 hour of usage users were presented with a web-based survey where they evaluated their experiences with the notification system. The results from the survey verified that 85% of the users were satisfied with the persona they selected, but many felt that they would prefer to be

able to customize the program explicitly. This fact prompted the addition of another user type to facilitate such an interaction.

4.2.5 Project Accomplishments

At an undergraduate research symposium for computer science students, NotiFly was selected overwhelmingly by over one hundred attendees for the “People’s Choice” award. In addition, NotiFly also received the 2nd place “Industry Choice” and “Dean’s Choice” awards.

5. DISCUSSION

In this section, we focus on some of the interesting design solutions devised by the NotiFly development team, as well as their possible implications for current HCI usability engineering theories.

5.1 The Emergence of Claims-Based Personas

The group decided to tackle the issue of system customization from a slightly different angle than typical scenario-based design. Normally, the designers would formulate claims for the various artifacts associated with a customizable interface and analyze their upsides and downsides with respect to critical parameters. Instead, the research group decided to approach customization with critical parameters through classes of users rooted in design claims—essentially, claims-based user personas.

These personas mitigated two usability engineering downsides in one fell swoop. First, the ongoing criticism of scenario-based design for its lack of understanding and empathy of potential users [1] is immediately combated with the addition of user personas. Lastly, the learning curve associated with customizing a system that a user is unfamiliar with can be eliminated by doing so behind the scenes based on selected user personas.

User personas were chosen from variations in critical parameters. It was assumed that when the user was alerted of an inexpensive flight ticket, their reaction would be the same—to purchase the ticket. Fixing the reaction critical parameter left four combinations of interruption and comprehension which yielded the four main user personas. As discussed previously, system testing by participants warranted the addition of a fifth user class that allowed for individual customization of interface elements.

The following is a listing and explanation of each user persona within the interface (see Figure 2).

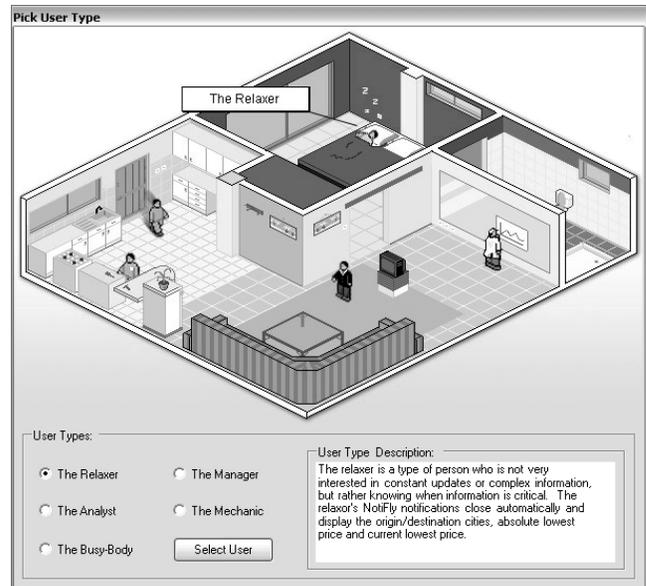


Figure 2. User persona selection screen of NotiFly

NotiFly User Personas:

- ***The Relaxer*** (Low I, Low C) – The relaxer user type is a user who does not want to be bothered with constant interruption or notification. They see their primary task as important and diversion from it as undesirable. They are furthermore unconcerned with long-term retention or understanding of secondary information.
- ***The Busy-Body*** (High I, Low C) – The busy-body user type is a user who is too engrossed in their primary task to constantly monitor notifications, but it is important that those notifications are received. This involves higher interruption and persistence of notifications to ensure important information is not missed. Like the relaxer, the busy-body is relatively unconcerned with secondary information.
- ***The Scientist*** (Low I, High C) – The scientist user type is a user whose primary focus is information and understanding. They desire a maximum amount of data in notifications to leverage comprehension, and also maintain an interest in formulating trends through remembering past notifications. Like the relaxer, the scientist sees their primary task as important and diversion from it as undesirable.
- ***The Manager*** (High I, High C) – The manager user type is a user who needs to know everything quickly. They need to make informed decisions in a timely fashion, so both receiving the notification and comprehending it is important to them.
- ***The Mechanic*** (Variable I, C) – The mechanic user type is a user who demands full control over their interface. This user type allows for complete customization of any interface element and represents a typical model for software customization.

5.2 Adaptable Notification System (ANS)

The most noticeable innovation that arose from the addition of user personas to the design model was the adaptable notification system (ANS). ANS is a notification system that custom-caters its delivery based solely upon the current user persona (see Figure 3). With this interface component, variables such as animation, color, duration or persistence of notification, and level of information delivered are set for the user automatically. This allowed the system to resonate with several different user models through only one design model—provided the user selected a persona that was most representative of their desired interaction.

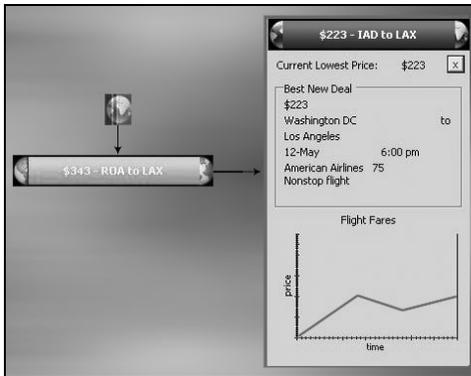


Figure 3. ANS in some of its various modes

5.2.1 ANS and Attentive Notification Systems

Discussion of ANS warrants mention of the domain of notification systems it resides in: attentive notification systems. Attentive notification systems deal with the process of adapting the delivery of information to avoid overloading the user [9]. The study of attentive theory is immediately beneficial under the IRC Framework, where interaction and comprehension are key foci.

ANS takes from the idea of an attentive notification system by adapting the delivery of the flight information based upon the user persona. The distinction between the two lies in that persona—attentive systems predict how the user wants the information delivered while the ANS bases that prediction off of the persona that the user selects themselves.

One of the problems with attentive notification systems is the cost associated with initially predicting the user's behavior. This is typically done by monitoring the user's actions over a set period of time and then changing the system's design model to close the gap between that model and the user's [9]. ANS mitigates this initial cost through personas, which allow the user to calibrate the system's design model according to their own perceived user model.

Future work on ANS would likely involve a merging of the two systems, utilizing user-selected personas to initially calibrate the design model and then monitoring the user's interactions and adapting that model attentively. Some of the ongoing research topics related to attentive user interfaces also lend themselves well to the refinement of adaptable notification systems by nature of their similarities. Some of these topics include:

- Systems built to adapt to individuals can mitigate problems associated with cognitive differences and interface learnability [9].

- Attentive user interfaces can detect changes in system goals and adapt design models accordingly [9].

6. CONCLUSIONS AND FUTURE WORK

The key findings in the development of NotiFly are summarized as follows:

- Considering user tasks over user goals, a frequent criticism of scenario-based design, is a problem diminished through the use of claims-based personas.
- Development of notification systems with critical parameters can greatly leverage the unification of design models in a multiple prototyping effort.
- The use of claims-based personas can lessen the initial calibration costs of attentive notification systems.

NotiFly was not only a successful undertaking in scenario-based design, but also a favorable indicator of potential future research. Developing multiple prototypes with a scenario-based approach, and then combining the best features and their design objects into one system allowed for a very rapid development cycle—which is counter-intuitive to what one might imagine. This leads us to believe that there is merit in investigating what advantages multiple prototyping can have to the entire design/development process. Additionally, the emergence of claims-based personas as a model for interface customization warrants further research to determine how one design model can be permuted into different user models dynamically. This structure could potentially alleviate the inherent interface bloating that occurs from accommodating a broad range of users in one system. Finally, the adaptive notification system contained in NotiFly draws interesting parallels to the world of attentive user interface (AUI) design, and further research in their harmony could lead to advances in both areas. Particularly, an attentive system that is allowed to “guess” adaptation through user-defined personas could make attentive systems more efficient. Likewise, an adaptive system could be made better through employing dynamic features found in AUIs.

7. REFERENCES

- [1] Carroll, J. M., Rosson, M. B., George, C. Jr., & Koenemann, J. “Requirements Development in Scenario-Based Design.” *IEEE Transactions on Software Engineering*. 24(12), 1998.
- [2] Chewar, C. M., Bachetti, E., McCrickard, D.S., & Booker, J. “Automating a Design Reuse Facility with Critical Parameters: Lessons Learned in Developing the LINK-UP System.” *Proc. Of the 2004 Intl Conf on Computer-Aided Design of User Interfaces (CADUI '04)*. 236-247, January 2003.
- [3] Cooper, A. *About Face 2.0: The Essentials of Interaction Design*. Wiley Publishing, Indianapolis, IN, 2003.
- [4] Goodwin, K. “Perfecting Your Personas.” *Cooper Interaction Design Newsletter*. July/August 2001.
- [5] McCrickard, D. S., Czerwinski, M., & Bartram, L. “Introduction: Design and Evaluation of Notification System Interfaces.” *International Journal of Human-Computer Studies* 8(5): 509-514, May 2003.

- [6] McCrickard, D. S., Chewar, C. M., Somervell, J.P., & Ndiwalana, A. "A Model for Notification Systems Evaluation--Assessing User Goals for Multitasking Activity." *ACM Transactions on Computer-Human Interaction (TOCHI)*, 10(4): 312-338, December 2003.
- [7] McCrickard, D. S. & Chewar, C. M. "Designing Attention-Centric Notification Systems: Five HCI Challenges." *Cognitive Systems: Human Cognitive Models in Systems Design*. Lawrence Earlbaum, 2005.
- [8] McCrickard, D. S., Chewar, C. M. "Links for a Human-Centered Science of Design: Integrated Design Knowledge Environments for a Software Development Process." To Appear *Hawaii Intl Conf on System Sciences (HICSS '05)*, Waikoloa, HI, January 2005.
- [9] McCrickard, D. S. & Chewar, C. M. "Attuning Notification Design to User Goals and Attention Costs." *Communications of the ACM*, 46(3):67-72, March 2003.
- [10] Rosson, M. B. & Carroll, J. M. *Usability Engineering: Scenario-Based Development of Human-Computer Interaction*. Morgan, San Francisco, CA, 2002.
- [11] Sutcliffe, A. G. & Carroll, J. M. "Designing Claims for Reuse in Interactive Systems Design." *Intl Journal of Human-Computer Studies* 50: 213-241, 1999.