Designing Attention-Centric Notification Systems: Five HCI Challenges

D. Scott McCrickard, C. M. Chewar
Department of Computer Science & Center for HCI
Virginia Tech
Blacksburg VA 24061-0106

Abstract: Through an examination of the emerging domain of cognitive systems, with a focus on attention-centric cognitive systems used for notification, this document explores the human-computer interaction challenges that must be addressed for successful interface design. This document asserts that with compatible tools and methods, user notification requirements and interface usability can be abstracted, expressed, and compared with critical parameter ratings; that is, even novice designers can assess attention cost factors to determine target parameter levels for new system development. With a general understanding of the user tasks supported by the notification system, a designer can access the repository of design knowledge for appropriate information and interaction design techniques (e.g., use of color, audio features, animation, screen size, transition of states, etc), which have analytically and empirically derived ratings. Furthermore, usability evaluation methods, provided to designers as part of the integrated system, are adaptable to specific combinations of targeted parameter levels. User testing results can be conveniently added back into the design knowledge repository and compared to target parameter levels to determine design success and build reusable HCI knowledge.

This approach is discussed in greater detail as we describe five HCI challenges relating to cognitive system development: (1) convenient access to basic research and guidelines, (2) requirements engineering methods for notification interfaces, (3) better and more usable predictive modeling for pre-attentive and dual-task interfaces, (4) standard empirical evaluation procedures for notification systems, and (5) conceptual frameworks for organizing reusable design and software components.

This document also describes our initial work toward building infrastructure to overcome these five challenges, focused on notification system development. We described LINK-UP, a design environment grounded on years of theory and method development within HCI, providing a mechanism to integrate interdisciplinary expertise from the cognitive systems research community. Claims allow convenient access to basic research and guidelines, while modules parallel a lifecycle development iteration and provide a process for requirements engineering guided by this basic research. The activities carried out through LINK-UP provide access to and interaction with reusable design components organized based on our framework. We think that this approach may provide the scientific basis necessary for exciting interdisciplinary advancement through many fields of design, with notification systems serving as an initial model.

A version of this document will appear as chapter 3 in the book *Cognitive Systems: Human Cognitive Models in Systems Design* edited by Chris Forsythe, Michael Bernard, and Timothy Goldsmith resulting from a workshop led by the editors in summer 2003. The authors are grateful for the input of the workshop organizers and conference attendees in the preparation of this document.

1. Introduction

Technological realities of multiple, ubiquitous information delivery streams for user notification often beg improved interface and usability in human-computer interaction (HCI). Many new HCI approaches hint at promising notification solutions, but the HCI field faces five important challenges that can be assisted by applied research in the cognitive systems community. When resolved, designers and researchers will have: convenient access to basic research and guidelines, requirements engineering methods for notification interfaces, better and more usable predictive modeling for pre-attentive and dual task interfaces, standard empirical evaluation procedures for notification systems, and conceptual frameworks for organizing reusable design and software components. This document provides an overview of these challenges and discusses some initial work undertaken in each area.

As we consider the general promise of adaptive interfaces, notification systems seem to be ideal for many situations. *Notification systems* are interfaces specifically designed to support user access to additional digital information from sources secondary to current activities (McCrickard, Czerwinski, & Bartram, 2003). Examples of notification systems include email alert devices, instant messengers, and in-vehicle information systems. When these systems are blended with technologies that can track and infer priorities of user attention and workload characteristics (such as through eye gaze, physical and biomedical sensors, and user models), *attentive user interfaces* (AUIs) result (Vertegaal & Velichkovsky, 1997). More specifically, when notification systems adapt information presentation and delivery to avoid overloading the user and to recommend content that may be of interest, we refer to these as *attention-centric systems* (Horvitz, 1999). The next section provides more information about user goals that relate to notification and introduces several examples of notification design artifacts and systems.

Reflecting on Forsythe's general vision of cognitive systems for interaction—where inferences from a user model or expert model provide opportunities for interaction that would expose users to alternate perspectives—we see roles for notification systems at each of the three levels. At the first level of this vision, the system acts as an *aide* that knows the user's priorities and interests and acts as a mediator of information. We already see examples of these types of systems emerging, such as the Scope notification system—an AUI that delivers alerts and provides an overview about incoming email, calendar tasks, and other information based on learned user priorities and expectations of urgency (van Dantzich, Robbins, Horvitz, & Czerwinski, 2002). If Forsythe's vision is to be realized in the years to come, we will see notification systems emerge that also act as *councils* and *oracles*. As a council, a notification system would be an interface for an expert agent that possesses unique domain knowledge and engages users transparently, using a variety of paradigms to conduct dialog-based and multimodal interaction. Oracles assume even more control over a user's implicit goals, seamlessly blending interaction responsibility among humans, agents, robots, and sensors. Success of notification systems that act as aides, councils, and oracles requires vast collections of empirically validated design artifacts, user models, and entity libraries. The challenges outlined in this document create the infrastructure necessary to accumulate such collections.

Our vision is part of a movement toward a science of design. In fall of 2003, the National Science Foundation (NSF) Directorate for Computer and Information Science and Engineering (CISE) sponsored a workshop toward the establishment of a Science of Design program, focusing on software-intensive systems. More than 50 prominent researchers, many from an HCI perspective, attended the workshop and each contributed a brief position paper. As HCI consumers of cognitive science research, we use this initiative to help frame the requirements for cognitive systems development. Our general approach is based on the recognition of a critical need to instantiate and operationalize a design method and environment that espouses hypothesis formation, testing, and iteration, co-evolution of problem specification and design solution, and progressive accumulation of design knowledge. Such an approach promises to inject a more rigorous, scientific approach to a discipline where it is lacking.

2. Background—User Notification Goals and Systems

The emerging discipline of notification systems provides an important area for the development of cognitive systems. Notification systems can be found in many implementation forms and on a variety of platforms (see Figure 1). Perhaps classic desktop systems are the most readily identifiable—instant messengers, status programs, and stock tickers. However, other familiar examples hint at the range of potential systems, such as Weiser's dangling string representation of network traffic (Weiser & Brown, 1996), in-vehicle information systems (Tufano, Knee, & Spelt, 1996), ambient media (Ishii et al, 1998), collaboration tools (Carroll et al, 2003), and multi-monitor displays (Grudin, 2001). Because notification systems are often lightweight tools (e.g., small peripheral displays in the corner of a desktop interface) that inform users about everyday information (e.g., airline ticket prices, news headlines, presence of collaborators or loved ones), designers are generally able to address important concerns with a relatively simple implementation effort. However, a user's initial acceptance and continued use of a notification system largely depends on satisfaction of their multitasking usage goals—leading to difficult design tradeoffs. Although many design efforts have emerged in recent years, techniques and methods for teaching, engineering, and evaluating usability for these types of interfaces have not been fully developed and evaluated—making this a design area ideal for focusing interdisciplinary research attention.

Early design efforts

In recent years, developers and researchers have demonstrated many innovative interface design approaches toward facilitating use of multiple simultaneous information sources. To introduce this research area, we provide examples of innovative systems and then discuss related work toward understanding psychological effects of these systems.

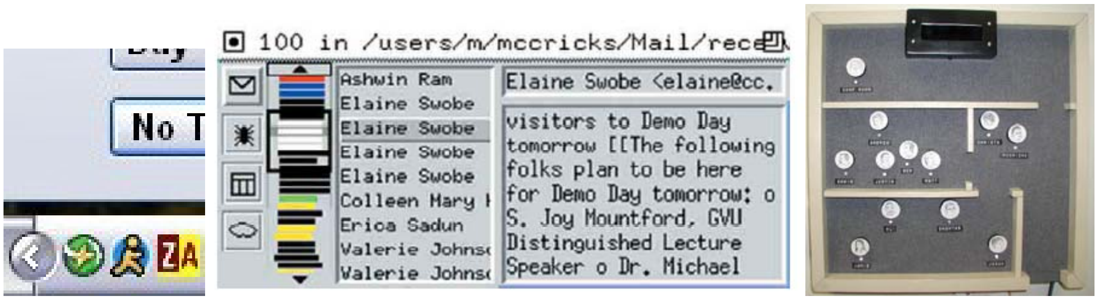

Figure 1: Desktop notification systems. From left to right: Notifly (Belcher et al., 2005) and other notification systems in the desktop toolbar; the desktop application Irwin (McCrickard, 1999); the off-the-desktop presence alert system Online Enlightenment (Heir et al., 2004).

Several efforts can be characterized as attempts to deliver information of interest with small desktop applications, specifically designed to provide glanceable awareness without disturbing other tasks or becoming annoying. The Scope (van Dantzich et al, 2002), Sideshow (Cadiz, Venolia, Jancke, & Gupta, 2002), and Irwin (McCrickard, 1999) applications adopt this strategy. As an alternative to dedicating constrained screen space to tickering displays and other notification tools, Harrison demonstrates transparent user interface objects, where overlaid notification information objects have some degree of transparency and can provide awareness of additional information and enhanced context with minimal obfuscation of other objects (Harrison, Ishii, Vicente, & Buxton, 1995). Other desktop notification applications do not seem to attempt to prevent distraction, instead proactively providing prompts intended to guide or enhance activities. Microsoft's Office Assistant (Clippit) and Rhodes and Maes' Remembrance Agent (2000) are examples.

Other innovative work has demonstrated feasibility and utility of presenting notifications 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 (MacIntyre et al, 2001) and efforts like Informative art (Redström, Skog, & Hallnäs, 2000), but there are fundamental differences in the objective amount of user attention necessary to extract information and gain meaning—sometimes there is great cost in terms of attention needed to process the information being displayed, whereas in other examples the information is subtly embedded in the environment. 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 (Dahley, Wisneski, & Ishii, 1998; Ishii et al, 1998; Heiner, Hudson, & Tanaka, 1999). Other work described how physical widgets can display information states with curious physical objects, such as an artificial flower arrangement (Greenberg & Fitchett, 2001). Although many of these examples are designed to enhance user efforts on desktop platforms, in classrooms, and in office environments (Mamykina, Mynatt, & Terry, 2001), similar research interest (and HCI expertise) often extends to cover more ubiquitous displays—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 that deliver information in divided-attention situations.

Information design studies

Understanding how to design aspects of these emerging systems has provided direction for several HCI research efforts. Some efforts within the community have focused on effective attention allocation by reducing distraction. Guidelines for in-vehicle information systems (IVIS) limit types of display interactions, restrict magnitude of display change, and display time (Bailey, Konstan, & Carlis, 2000; Green, 1999; Tufano et al, 1996). Other approaches have sought to optimize selection of attention demands by considering associated cost of user interruption and appropriately tailoring notification presentation (Maglio & Campbell, 2000). Horvitz's inference procedures for automated notification systems are one example—driven by his belief that human attention is the most valuable commodity in HCI (Horvitz, 1999; Horvitz, Jacobs, & Hovel, 1999; Horvitz, Kadie, Paek, & Hovel, 2003). Similarly, McFarlane describes a taxonomy and empirical study of the major dimensions and design tradeoffs related to interruption coordination methods (McFarlane, 1998; McFarlane, 2002).

In contrast to the approach of preserving attention as best as possible, other researchers have focused on optimizing the presentation of notification items that provide users some type of measurable utility (Wickens & Hollands, 2002). Empirical evaluations often simulate dual-task situations, asking participants to perform a primary task while reacting to secondary displays and measuring utility (benefits from acquiring the information) and costs (e.g., in task correctness and completion time). 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 (Healey, Booth, & Enns, 1996; Healey & Enns, 1999; Bartram, Ware, & Calvert, 2001). Earlier work examined moving and changing text as a method to present information, observing the perceptibility and readability of rapid serial visual presentations (RSVPs) of letters, strings, and words (Duchnicky & Kolers, 1983). Rather than optimizing displays for quick reaction, another approach has been to increase utility with information design options that allow deeper understanding and memorability (Kang & Muter, 1989). For example, Cutrell, Czerwinski, and Horvitz investigated impacts of messaging on primary task related memory and performance (2001).

There may be some hope that these various efforts present diverse approaches to filling user needs, perhaps eventually converging to provide complete coverage of design challenges. However, there are no known mechanisms in place to facilitate collection and analysis of this design advice, either in the form of guidelines, tradeoffs, or system examples. Furthermore, the field lacks widely accepted usability engineering and evaluation processes that could be integrated into HCI education and cohesive research efforts.

An Underlying Conceptual Model for Notification Design

Considering the commonalities in the systems like those introduced earlier, we have recognized a few general goals. In McCrickard, Catrambone, Chewar, & Stasko (2003), we noted an important distinction between notification systems and traditional HCI research, the attention-utility theme, asserting that it is useful to think of attention as a constrained resource that can be traded for utility. This utility is enabled by perceiving additional, valued information while performing other tasks: *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). In (McCrickard & Chewar, 2003), we summarize attention benefits (e.g., understanding patterns, trends, and changes, providing responses, prompting task transition, etc.) and situational cost factors that notification system users generally expect.

IRC Ratings: Critical Parameters for Usability Engineering Decisions. Users ultimately use a notification system to gain benefits, which come from specific types of utility, which can result from associated user goals. We recognize that the three general user goals of comprehension, reaction, and interruption can be thought of as critical parameters—key measures of system success that can be benchmarked through empirical testing to reveal design progress (McCrickard, Chewar, Somervell, & Ndiwalana, 2003). These goals are unique in that the user is willing to sacrifice a certain amount of primary task attention to achieve them. Other important system features and user needs must be typically supported in user interfaces, including 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 situational factors (fully detailed in (McCrickard & Chewar, 2003)).

With compatible tools and methods, user notification requirements and/or interface usability can be abstracted, expressed, and compared with three parameter ratings (an interruption/reaction/comprehension (IRC) rating)—that is, designers without dual-task analysis expertise can assess attention cost factors to determine target IRC levels for a new notification system. Factors such as a user's lack of skill in perceiving unfamiliar or complex notification information may contribute to these parameters, and objective ratings may not carry a constant value across different situations. With this rating and a general understanding of the user tasks supported by the notification system, a designer can access the repository of design knowledge for appropriate information and interaction design techniques (e.g., use of color, audio features, animation, screen size, transition of states, etc.)—which have analytically and empirically derived IRC ratings. Furthermore, usability evaluation methods, provided to designers as part of the integrated system, are adaptable to specific combinations of targeted parameter levels. User testing results can be conveniently added back into the design knowledge repository and compared to target parameter levels to determine design success and build reusable HCI knowledge. This approach is discussed in greater detail as we describe five HCI challenges relating to cognitive system development.

3. The Five HCI Challenges

Improve access to basic research and guidelines

Certainly the challenge of designing effective notification systems is formidable when considering the range of possible psychological effects for diverse groups of users. As we reviewed earlier, researchers of basic psychological questions are making progress in understanding information and interaction techniques that are effective for dual task situations. Unfortunately, these results tend to be outside the reach of ordinary interface designers as they lack the disciplinary background to decipher published articles in cognitive and experimental psychology and human factors. High-level summaries that might deliver useful guidelines tend not to be available.

Challenge #1: The HCI development community must have convenient access to basic research and guidelines for attention-centric notification design.

One of the fundamental goals of human-centric design is to equip interface developers with techniques that allow them to design more usable systems. Part of this goal is accomplished when designers can recognize points in the design cycle and user interactive experience that can benefit from mitigating psychological effects on the user. However, this goal is only fully accomplished when designers have at their disposal a variety of options that are informed by results from empirical testing and grounded in widely accepted theory. As researchers from psychology, human factors, and HCI, when we are unable to deliver such option sets to designers, we fail to provide a critical service to a key information consumer.

As the general software development community looks at the work that must be done to improve our practice as a science of design, we see many nuances contributing to the challenge that notification systems designers face. For example, Bonnie John surfaces many related considerations stemming from breakdowns in design teamwork that necessitate focus within a Science of Design program (John, 2003). On the premise that the software development community must provide nearly all of the infrastructure required for the interdisciplinary transfer of knowledge, she argues that significant research must be sponsored—to include process and tool support for enhancing contributions from interdisciplinary team members—helping to deliver knowledge from the behavioral sciences for prediction of design idea feasibility prior to extensive building or prototyping efforts.

Other observations from prominent researchers 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. Jakob Nielsen, a principal of the Nielsen Norman Group and author of several influential books on usability engineering, 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 (Nielsen, 2003). These new business practices imply that the challenge of delivering basic research results to system developers will become increasingly more difficult in years to come.

Deliver requirements-engineering methods

The second challenge addresses the core design process for software interfaces. Many contemporary approaches assist designers in applying design guidelines, and some of the more promising methods encourage iterative refinement through integrated testing and analysis. However, most people approach design as a purely creative process, lacking in structure and documentation, with the only product being the designed artifact itself. If we are to make progress toward Forsythe's vision and move toward attention-centric notification systems, we must look for processes that encourage designers to form and test hypotheses, preserving the knowledge gained through a channeled creative process.

Challenge #2: Processes and frameworks should be available for requirements engineering and development of interaction specifications for software engineers.

Several prominent researchers provide thoughts on new requirements for methods that would center on three important themes that will enable the full potential of cognitive systems to be realized. The first theme is an argument for design methods that support an improved ***understanding of problem spaces***. As noted by Turing Award winner Fred Brooks, "often the hardest part of design is deciding *what* to design," because designers often lack a precise description of the problem to be solved (Brooks, 2003). Long-time software developer and AT&T researcher Michael Jackson elaborates: "a science of design must be at root a discipline of devising, understanding, populating, and exploiting [an] informal structure" so that software development problems can be decomposed into sub-problems within known problem frames (Jackson, 2003).

Brooks also introduces a second theme for software development processes that embodies the movement to increase the scientific basis of design—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 (Brooks, 2003). Several other prominent researchers echo and elaborate this sentiment. Warnier-prize winner Mary Shaw, co-director of the Sloane Software Engineering Center and author of seven books, argues for systematic guidance of design decisions, specifically those that express costs and benefits of software design and help designers consider user preferences (Shaw, 2003). CHI Lifetime Achievement Award winner John M. Carroll notes that this knowledge must focus on user activities in a way that leverages research from social sciences (Carroll, 2003). Colin Potts argues that knowledge should be accumulated by recording the "science of the designed" through artifact-as-phenomena investigations, modeled as pattern abstraction (Potts, 2003). These ideas are summarized in the second challenge for the cognitive systems research community as we strive to lay the groundwork for attention-centric notification systems.

Enhance and expand predictive modeling capabilities

As interface developers and HCI researchers consider ways to lower costs of software development yet developing highly usable systems, cognitive modeling may be compelling. Rather than invest in costly user testing, HCI professionals are often intrigued at the idea of discovering breakdowns in human information processing through automated methods. With cognitive architectures like ACT-R (Anderson, 1998), SOAR (Lewis, 1999), and EPIC (Kieras, Wood, & Meyer, 1997), which have been in development and appeared in the literature since the early 1990s, we have reason to hope that these architectures should provide design requirements insight for typical designers. Unfortunately, the systems based on the architectures currently available do not come close to meeting the needs of most notification system designers and researchers.

Challenge #3: Better and more usable predictive modeling for pre-attentive and dual-task interfaces must be available to HCI researchers and system developers.

Cognitive modeling has been successfully used to investigate specific design-related questions and support theory. One example probed typical user behavior with online help manuals (Peck & John, 1992), and studies of dual-task performance in driving situations with in-vehicle systems are fairly common (e.g., Salvucci, 2001). However, for these models to be effective, they must specify mechanisms and constraints related to human information processing and graphical user interface interactivity, such as limitations of human memory, attention, facilities, perceptual-motor operation characteristics, response selection limitations, and sensory perception performance abilities.

Constructing models that encapsulate these vast collections of theories and results of empirical studies is extremely complex and often forces models to be created to address narrow problems such as those introduced earlier. Not only do the problem domains that a given model may address tend to be narrow, but models also tend to be extremely difficult to configure and use for problem solving. Designers who are unfamiliar with the model face challenges in deciding whether the problem they are investigating can even be validly studied using the model. These are some of the key findings that resulted from our own study, in which a novice designer attempted to use three cognitive architectures to probe questions relating to notification system design (Turnbull, Chewar, & McCrickard, 2003). Although each of the three architectures exhibited usability and performance-positive features, none was able to support a novice designer's requirement to quickly learn and sufficiently customize a predictive model for a simple information design question that would be typical in early-stage design.

Standardize empirical evaluation procedures

Although use of notification systems has become widespread in recent years, there are surprisingly few efforts within HCI literature that effectively evaluate usability of their information and interaction design. For example, some notification systems support collaborative activities and are studied from a computer-supported collaborative work (CSCW) perspective, whereas disparate agendas lead to inconsistent definitions of successful design, inhibiting cross-initiative influence. Perhaps a leading cause for the general lack of user studies in reporting system development efforts and innovative design solutions is the lack of training that software developers typically receive in designing experiments. Although some developers make the extra effort to test systems and report findings, this is often done with procedures that are not replicable or comparable in related efforts. This practice prevents long-term research growth and disciplinary cohesion. However, by helping software developers create reusable test platforms and instruments that can capture key usability concerns, researchers from disciplines that value empirical data collection can contribute to our long-term success.

Challenge #4: Usability engineers need assistance in developing standard and reusable evaluation procedures for notification system interfaces.

As one of the two important research challenges asserted by Abowd and Mynatt (2000) for the emerging interfaces for computing, they motivate the imperative for assessing progress toward real human needs with quantitative and qualitative evaluation methods that capture authentic context of system use, saying that research irrespective of the need for evaluation will have little impact in the HCI community. In response, this general message has guided the agenda of two workshops at major HCI conferences and special issues of journals. Within the notification systems research community, there has been encouraging momentum with the development of generic usability evaluation tools, especially for specific classes of systems. Heuristic evaluation methods have been adapted for both ambient displays (Mankoff et al, 2003) and large-screen information exhibits (Somervell, Wahid, & McCrickard, 2003). The key benefits of generic evaluation tools include the ability to easily compare and benchmark system performance, recognize progress toward reference tasks, and collect experience necessary for cost-benefit reengineering assessments (Whittaker, Terveen, & Nardi, 2000). Although there may be some concern that generic evaluation tools are not capable of providing rich and expressive results about a particular set of features, we have obtained results to the contrary in probing this general hypothesis (Somervell, Chewar, McCrickard, & Ndiwalana, 2003).

Provide conceptual frameworks for design reuse

The ability to reuse components from one design to the next represents cost savings in the form of both development efficiency and improved reliability for usability. However, the benefits of reuse are often only available after a practice has reached a sufficient level of maturity, apparent through establishment and acceptance of formal methods. With reusable components that have understood psychological effects on users, notification systems designers are able to create adaptive interfaces appropriately tailoring presentation to accommodate the needs for interruption, reaction, and comprehension (McCrickard & Chewar, 2003). This adaptivity requires a rich set of reusable components that are well organized according to critical user concerns. Unfortunately, we have seen little indication that reuse even occurs on a system-to-system basis, a state that would allow designers to improve on the work of their predecessors and perhaps an achievable near-term goal on which to focus.

Challenge #5: Conceptual frameworks must be crafted to assist in organizing reusable design knowledge and software components—a necessity for design efficiency and long-term progress.

As we consider arguments made by prominent researchers that support the goal of improving designer tendencies to reuse design knowledge, two fundamental concepts provide a possible starting point. First, Carroll (2000) described how reusable statements about the psychological effects of a design artifact in use (claims within a claims analysis) can act as a hillclimbing heuristic. By *hillclimbing*, he meant 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. 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 (Carroll, 2000). In the next section, we extend this concept to include co-evolutionary development of both the design and user's model.

As a second fundamental concept, we consider William Newman's notion of critical parameters**,** which provides us with a mechanism to measure our progress in hillclimbing. To conduct meaningful modeling and usability evaluations that allow systems to become progressively better, and in response to his 1994 study of CHI paper contributions (Newman, 1994), Newman argued 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 (Newman, 1997; Newman, Taylor, Dance, & Taylor, 2000). He implied 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. Whittaker et al. (2000) extended his arguments to a proposed reference task agenda for HCI to increase community scientific focus. The convergence of these ideas provides the theoretical basis for a potential solution that we will proceed to discuss: the iterative process of gauging critical parameters, embodied in design artifacts and expressed with claims, guides the hillclimbing process and provides an index for archived, reusable design knowledge.

4. Toward a Solution to LINK-UP Our New Discipline

Although there are many potential approaches for injecting scientific inquiry into HCI and the interface design process (e.g., Hix & Hartson, 1993), we augment the task-artifact framework, embedded in an iterative scenario-based design process, facilitating design knowledge hypothesis formation, tradeoff mitigation, and component reuse. We are developing a design environment specifically to assist novice developers with analyzing and constructing design rationale for notification systems. The IRC framework described earlier in the document provides the theoretical and technical underpinnings for the tools we are building and integrating.

Task-Artifact Framework and Usability Engineering. In the late 1980s, Carroll introduced a proposal for a systematic method to reconcile contrasting perspectives of hermeneutics and theory-based design (Carroll & Kellogg, 1989). 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 (Carroll, Singley, & Rosson, 1992).

In the description of this process, Carroll noted 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. In later work, Carroll argued 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 (Carroll & Rosson, 1992). Based on the task-artifact framework, Carroll developed a gradient of progressively powerful analysis techniques, starting with basic scenario-based design and task coverage through Norman's stages of action, and extending to the process of claims analysis and hillclimbing, 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 (Carroll, Mack, Robertson, & Rosson, 1994)). We review the basic techniques, which form our co-evolutionary design method.

Norman's stages of action and conceptual models. One classic theory in interface design literature is Norman's (1986) theory of action. Because user tasks are composed of psychological goals and intentions and are accomplished with control mechanisms to physically manipulate system states, he recognized two different expressions of a task (physical and psychological) that must be resolved within an HCI system. Norman established the idea that governing the usage experience is the consistency of two conceptual models—the *design model* held by the designer and the *user's model* based on the user's understanding of the system. Each model can be analyzed as stages of action, which describe the cyclical evaluation and execution of tasks across the *Gulf of Execution* and the *Gulf of Evaluation.* 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, including 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 (SBD) 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. Through the development and sharing of narrative descriptions of users solving problems with designed systems, designers are able to create the scaffolding across Norman's Gulfs—and develop systems with design-user's model compatibility. Enabling designers to compare these conceptual models, as well as research and improve suitable design artifacts, is a central goal of our work.

Claims Analysis. 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, developers 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 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" (Carroll, 1994).

Related work by Sutcliffe has developed theories and methods for design reuse in the requirements generation stage (Sutcliffe, 2000). Based on his work with Carroll, he argued that HCI research should focus on producing "designer digestible" packets of HCI knowledge in the form of claims, grounded on solid theory and allowing general reuse (Sutcliffe & Carroll, 1999). To this end, Sutcliffe's Domain Theory provides a structure of abstraction, formal definitions, reuse program evaluation metrics, and generic tasks that can be used to catalogue design information—an implementation roadmap extendible to any design domain that is employed in plans for our design tool.

A Notification Systems Claims Library. We designed and implemented a claims library for notification systems artifacts and design knowledge, which uses Domain Theory (Sutcliffe, 2002) components and IRC rating framework (McCrickard, Chewar, Somervell, & Ndiwalana, 2003) as an index. The claims library serves as an underlying component for the system, where an example of a claim in simplest form could be:

Use of **tickering text-based animation** to display news headlines in a small desktop window:

+ Preserves user focus on a primary task, while allowing long-term awareness
BUT (-) is not suitable for rapid recognition of and reaction to urgent information.

Claims are grounded by empirical testing or observation, so a designer of a notification system may compare this claim with claims related to use of in-place animation techniques, such as fading and blasting. To simplify the process, we can abstract a claim and focus on critical parameters relating information presentation to effects on information processing (IRC)—user's interruption, reaction, and comprehension:

Tickering text-based animation $\in$ {*low* interruption, *low* reaction, *moderate* comprehension}

Full claims records that are stored in our design knowledge repository can be quite detailed, so appropriate generalization practices and search tools must adequately support an abstraction-specification process. Other publications from our group detail the approaches and findings in more depth (Payne et al., 2003; Chewar, Bachetti, McCrickard, & Booker, 2004; Wahid, Allgood, Chewar, McCrickard, 2004). Figure 2 details some of the key stages in the LINK-UP system, described in the rest of this section.

Requirements Analysis—Understanding the Design Model. 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. 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, hierarchical task analysis, matching of requirements to standard task models, and decomposing a task model to stages of action.

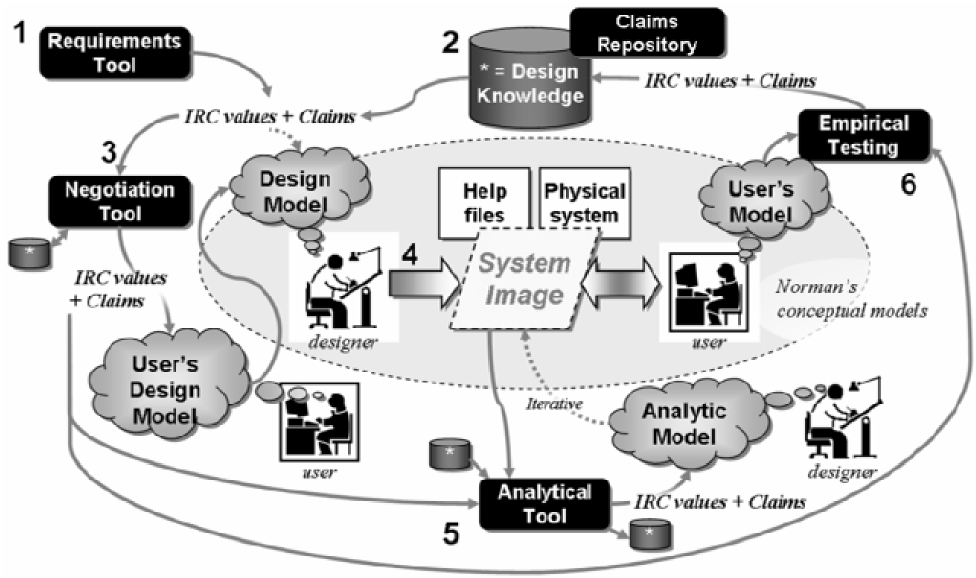

Figure 2: Key architectural components of LINK-UP. The light grey center region depicts Norman's conceptual models, extended through our work as described in this section. (Figure from Chewar, Bachetti, McCrickard, & Booker, 2004.)

Enhancing claim development and reuse through participatory negotiation. When designers sit down with users in a participatory design session, there are many possible negotiation points. In our work, we focus on scenarios, claims, and hierarchical task analysis diagrams, combining participatory design with the LINK-UP system—thus enabling reuse of claims to be accomplished in a participatory negotiation session.

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, which emerges from the requirements analysis module, this module facilitates designers in presenting their understanding of requirements to stakeholders and receiving specific feedback. Tools within this module allow a designer to build a participatory negotiation session and allow a stakeholder to take part in one.

Expressing and analyzing a system image with claims. After a designer develops a system image (either through the full implementation of a working system or a minimally functional prototype), this tool allows description of the key interface features with design claims. Revisiting the stage of action template holding the problem claims, the designer links each problem claim to a scenario and several design claims (information or interaction). 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 within this module 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.

Automatically creating empirical tests for claims evaluation. As designers proceed through the design process, they generate a huge collection of claims. Sometimes individual claims include empirical supporting evidence, but often they do not, and rarely are all sets of associated claims in the interface adequately validated. This presents an issue in designing and developing empirical tests to investigate the quality of the interface.

Designers and evaluators want to select sets of claims most appropriate and desired for user testing. Claims are grouped within and sequenced between the different stages of action. This module helps with the selection of information and interaction usage scenarios, creation of a test script as an input file for an automated test platform, and generation of an output file from the platform that could then append a claim record with the derived parameter values. The testing procedures supported by the tool are compatible with literature on critical parameters, reference tasks, and standard lab-based procedures for interface testing.

Visualizing claim linkages in LINK-UP. This module visualizes the changes an interface has gone through while being developed. The visualization specifically depicts the claims and scenarios that were used for an interface over time, showing the evolution of claims and scenarios and the relationships that are formed and working closely with the other modules.

Integrated Design Knowledge Reuse. A main advantage of the LINK-UP system is that it provides continuous and integrated access to the design knowledge repository, facilitating knowledge reuse. The design knowledge repository will build directly on a working prototype system that is based on Sutcliffe's Domain Theory (2000). Through access to the claims database, designers will be able to build from and test previous design results. They will also be able to contribute to a growing body of knowledge. To enable these features in a manner that preserves content quality and user trust, the system also includes accounts and profiles of designers and expert administrators. Expert access to the claims database allows full claims administration, association of claims with related theories, example systems, design artifacts, and other meta-analysis and knowledge management features, such as a claims entry, editing, rating, and commenting features for designers.

System Summary. In summary, 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.

5. Looking to the Future

Within this document, we have described a research area worthy of sustained interest from the cognitive systems community—attention-centric notification systems. As a research area, the study of notification systems can act as an incubator for the development of Forsythe's vision about the levels of cognitive system maturity. To focus this development in a way that will assist the software developer in making useful and usable systems for a wide diversity of users, we have introduced five challenges to guide research and practical exploration between HCI and cognitive systems research: (1) convenient access to basic research and guidelines, (2) requirements engineering methods for notification interfaces, (3) better and more usable predictive modeling for pre-attentive and dual-task interfaces, (4) standard empirical evaluation procedures for notification systems, and (5) conceptual frameworks for organizing reusable design and software components.

We also describe our initial work toward building infrastructure to overcome these five challenges, focusing on notification system development. We described LINK-UP, a design environment grounded on years of theory and method development within HCI—providing a mechanism to integrate interdisciplinary expertise from the cognitive systems research community. Claims and the claims library act as a repository that allows convenient access to basic research and guidelines, while the modules parallel a lifecycle development iteration and provide a process for requirements engineering guided by this basic research. We are already integrating standard empirical evaluation tools, extensible so that as we and others develop usable cognitive architectures they can be integrated as well. The activities carried out through the LINK-UP system provide access to and interaction with reusable design components organized based on our theoretical framework. We think that approaches like this may provide the scientific basis necessary for exciting interdisciplinary advancement through many fields of design, with notification systems serving as an initial model.

References

Abowd, G. D. & Mynatt, E. D. (2000). Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction*, *7*, 29-58.

Anderson, J. (1998). *The Atomic Components of Thought*. Mahwah, NJ. Erlbaum.

Bailey, B. P., Konstan, J. A., & Carlis, J. V. (2000). 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)*, Tokyo, Japan, 593–601.

Bartram, L., Ware, C., & Calvert, T. (2001). Moving icons: Detection and distraction. In *Proceedings of the IFIP TC.13 International Conference on Human-Computer Interaction (INTERACT 2001)*, Tokyo, Japan, 57-165.

Belcher, J., Aidrus, R., Congleton, B., Hall, D., Hussain, S., Jablonski, M., and McCrickard, D. S. (2005). Notifly: Enhancing design through claims-based personas and knowledge reuse. In *Proceedings of the ACM Southeast Conference (ACMSE 2005),* Kennesaw GA, 2-376-2-381.

Brooks, F. (2003). Is there a design of design? Position paper for the *NSF Invitational Workshsop on Science of Design: Software-Intensive Systems*, Airlie Center, VA.

Cadiz, J. J., Venolia, G., Jancke, G., and Gupta, A. (2002). Designing and deploying an information awareness interface. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*, New Orleans LA, 314–323.

Carroll, J. M. (1994). Making use: a design representation. *Communications of the ACM* 37 (12), 28-35.

Carroll, J. M. (2000). *Making Use: Scenario-Based Design of Human-Computer Interactions*. The MIT Press, Cambridge, MA.

Carroll, J. M. (2003). Activity as the object of design. Position paper for the *NSF Invitational Workshsop on Science of Design: Software-Intensive Systems*, Airlie Center, VA.

Carroll, J. M. and Kellogg, W. A. (1989). Artifact as theory-nexus: hermeneutics meets theory-based design. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 1989)*, Austin TX, 7-14.

Carroll, J. M., Mack, R.L., Robertson, S., Rosson, M. B. (1994). Binding object to scenarios of use. *International Journal of Human-Computer Studies 22*(2), 243-276.

Carroll, J. M. and Rosson, M. B. (1992). Getting around the task-artifact cycle: how to make claims and design by scenario. *ACM Transactions on Information Systems (TOIS) 10* (2), 181-202.

Carroll, J. M., Singley, M. K., and Rosson, M. B. (1992). Integrating theory development with design evaluation. *Behavior and Information Technology 11*, 247-255.

Carroll, J. M., Neale, D. C., Isenhour, P. I., Rosson, M. B., and McCrickard, D. S. (2003). Notification and awareness: Synchronizing task-oriented collaborative activity. *International Journal of Human-Computer Studies, 8*(5):605-632.

Chewar, C. M., Bachetti, E., McCrickard, D. S., and Booker, J. E. (2004). 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 2004),* Island of Mediera, Portugal, 236-247.

Cutrell, E., Czerwinski, M., and Horvitz, E. (2001). 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)*, Tokyo, Japan, 263–269.

Dahley, A., Wisneski, C., and Ishii, H. (1998). Water lamp and pinwheels: Ambient projection of digital information into architectural space. In *Conference Companion of the ACM Conference on Human Factors in Computing Systems*, Los Angeles CA, 269–270.

Duchnicky, R. L., and Kolers, P. A. (1983). Readability of text scrolled on visual display terminals as a function of window size. *Human Factors 25*(6):683–692.

Green, P. (1999). The 15-second rule for driver information systems. In *Proceedings of the ITS America Ninth Annual Meeting*, Washington, DC, CD–ROM.

Greenberg, S., and Fitchett, C. (2001). 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, 209-218.

Grudin, J. (2001). Partitioning Digital Worlds: Focal and Peripheral Awareness in Multiple Monitor Use. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '01)*, Seattle, WA, 458-465.

Harrison, B. L., Ishii, H., Vicente, K. J., and Buxton, W. A. S. (1995). Transparent layered user interfaces: An evaluation of a display design to enhance focused and divided attention. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '95)*, Denver CO, 317-324.

Healey, C. G., Booth, K. S., and Enns, J. T. (1996). High-speed visual estimation using preattentive processing. *ACM Transactions on Human Computer Interaction*, *3*(2):107–135.

Healey, C. G., and Enns, J. T. (1999). Large datasets at a glance: Combining textures and colors in scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, *5*(2):145–167.

Heiner, J. M., Hudson, S. E., and Tanaka, K. (1999). The information percolator: Ambient information display in a decorative object. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '99)*, Asheville, NC, 141-148.

Heir, M., Hoon, H., Terrell, G., and McCrickard, D. S. (2004). Online Enlightenment: APhidget Notification System for Online Status. Technical Report TR-04-30, Computer Science, Virginia Tech.

Hix, D. and Hartson, H. R. (1993). *Developing User Interfaces: Ensuring Usability Through Product and Process*. John Wiley and sons, New York, NY.

Horvitz, E. (1999). Principles of mixed-initiative user interfaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '99)*, Pittsburgh, PA, 159-166.

Horvitz, E., Jacobs, A., and Hovel, D. (1999). Attention-sensitive alerting. In *Conference on Uncertainty and Artificial Intelligence (UAI '99)*, Stockholm, Sweden, 305-313.

Horvitz, E., Kadie, C., Paek, T., and Hovel, D. (2003). Models of attention in computing and communication: from principles to applications. *Communications of the ACM*, *46*(3):52–59.

Ishii, H., Wisneski, C., Brave, S., Dahley, A., Gorbet, M., Ullmer, B., and Yarin, P. (1998). ambientROOM: Integrating ambient media with architectural space. In *Conference Companion of the ACM Conference on Human Factors in Computing Systems (CHI '98)*, Los Angeles CA, 173-174.

Jackson, M. (2003). Position paper: A science of software design? Position paper for the *NSF Invitational Workshsop on Science of Design: Software-Intensive Systems*, Airlie Center, VA.

John, B. E. (2003). Position statement of Bonnie E. John for Science of Design: Software-Intensive Systems. Position paper for the *NSF Invitational Workshsop on Science of Design: Software-Intensive Systems*, Airlie Center, Virginia.

Kang, T. J. and Muter, P. (1989). Reading dynamically displayed text. *Behaviour and Information Technology*, *8*(1):33–42.

Kieras, D. E., Wood, S. D., and Meyer, D. E. (1997). Predictive engineering models based on the EPIC architecture for a multimodal high-performance human computer interaction task. *ACM Transactions on Human Computer Interaction 4*, 3, 230-275.

Lewis, R. L. (1999). Cognitive modeling, symbolic. In *Wilson, R. and Keil, F. (eds), The MIT Encyclopedia of the Cognitive Sciences.* Cambridge, MA: MIT Press.

MacIntyre, B., Mynatt, E. D., Voida, S., Hansen, K. M., Tullio, J., and Corso, G. M.. 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)*, Orlando FL, 41-50.

Maglio, P. P. and Campbell, C. S. (2000). Tradeoffs in displaying peripheral information. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2000)*, The Hague, The Netherlands, 241-248.

Mamykina, L., Mynatt, E. D., and Terry, M. A. (2001). Time aura: Interfaces for pacing. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*, Seattle WA, 144-151.

Mankoff, J., Dey, A., Hsieh, G., Kientz, J., Lederer, S., and Ames, M. (2003). Heuristic evaluation of ambient displays. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI '03)*, ACM Press, 167-176.

McCrickard, D. S. (1999). Maintaining information awareness with Irwin. In Proceedings of the World Conference on Educational Multimedia/Hypermedia and Educational Telecommunications (ED-MEDIA '99), Seattle, WA, CD-ROM.

McCrickard, D. S., Catrambone, R., Chewar, C. M., and Stasko, J. T. (2003). Establishing tradeoffs that leverage attention for utility: Empirically evaluating information display in notification systems. *International Journal of Human-Computer Studies 8*(5): 547-582.

McCrickard, D. S. and Chewar, C. M. (2003). Attuning notification design to user goals and attention costs. *Communications of the ACM 46*(3):67–72.

McCrickard, D. S., Chewar, C. M., Somervell, J. P., and Ndiwalana, A. (2003). A model for notification systems evaluation—Assessing user goals for multitasking activity. *ACM Transactions on Computer-Human Interaction (TOCHI) 10*(4), 312-338.

McCrickard, D. S., Czerwinski, M., and Bartram, L. (2003). Introduction: Design and Evaluation of Notification User Interfaces. *International Journal of Human-Computer Studies 8*(5): 509-514.

McFarlane, D. C. (1998). *Interruption of People in Human-Computer Interaction*. PhD thesis, George Washington University,Washington DC.

McFarlane, D. C. (2002). Comparision of four primary methods for coordinating the interruption of people in human-computer interaction. *Human Computer Interaction*, *17*(3).

Newman, W. M. (1994). A preliminary analysis of the products of HCI research, using pro forma abstracts. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '94)*, Boston MA, 278-284.

Newman, W. M. (1997). 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)*, ACM Press, 239-245.

Newman, W. M., Taylor, A. S., Dance, C. R., and Taylor, S. A. (2000). 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)*, ACM Press, 381-387.

Nielsen, J. (2003). Getting the science of design used. Position paper for the *NSF Invitational Workshsop on Science of Design: Software-Intensive Systems*, Airlie Center, Virginia.

Norman, D. A. (1986). Cognitive engineering. In Donald A. Norman and Stephen W. Draper, editors, *User Centered System Design: New Perspectives on Human Computer Interaction*, Lawrence Erlbaum Associates, 31-62.

Payne, C., Allgood, C. F., Chewar, C. M., Holbrook, C., and McCrickard, D. S. (2003). 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 2003),* Las Vegas NV, 362-369.

Peck, V. A. and John, B. E. (1992). Browser-Soar: A computational model of a highly interactive task. In *Proceedings of the Conference of Human Factors in Computing Systems (CHI 1992)*, Monterey CA, 165-172.

Potts, C. (2003). Policy modularity: Toward a science of socially-embedded system design. Position paper for the *NSF Invitational Workshsop on Science of Design: Software-Intensive Systems*, Airlie Center, VA.

Redström, J., Skog, T., and Hallnäs, L. (2000). Informative art: Using amplified artworks as information displays. In *Proceedings of Designing Augmented Reality Environments (DARE 2000)*, 103-114.

Rhodes, B. and Maes, P. (2000). Just-in-time information retrieval agents. *IBM Systems Journal*, *39*(3–4):685–704.

Salvucci, D. D. (2001). Predicting the effects of in-car interface use on driver performance: An integrated model approach. *International Journal of Human-Computer Studies 55,* 85-107.

Shaw, M. (2003). Research questions for foundations of complex software design. Position paper for the *NSF Invitational Workshsop on Science of Design: Software-Intensive Systems*, Airlie Center, VA.

Somervell, J., Chewar, C. M., McCrickard, D. S., and Ndiwalana, A. (2003). Enlarging usability for ubiquitous displays. In *Proceeding of the 41st Annual ACM Southeast Conference (ACM-SE 2003)*, Savannah GA, 24–29.

Somervell, J., Wahid, S., and McCrickard, D. S. (2003). Usability Heuristics for Large Screen Information Exhibits. Short paper in *Proceedings of the IFIP TC.13 Conference on Human-Computer Interaction (INTERACT 2001)*, Zurich Switzerland, 904-907.

Sutcliffe, A. (2000). On the effective use and reuse of HCI knowledge. *ACM Transactions on Computer-Human Interaction*, *7*(2):197–221.

Sutcliffe, A. (2002). *The Domain Theory: Patterns for Knowledge and Software Reuse*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ.

Sutcliffe, A. G. and Carroll, J. M. (1999). Designing claims for reuse in interactive systems design. *International Journal of Human-Comupter Studies 50* (3), 213-241.

Turnbull, D. G., Chewar, C. M., & McCrickard, D. S. (2003). Are cognitive architectures mature enough to evaluate notification systems?" In *Proceedings of the International Conference on Software Engineering Research and Practice (SERP 2003)*, Las Vegas NV, 5 pages.

Tufano, D., Knee, H., and Spelt, P. (1996). In-vehicle signing functions and systems concepts. In *Proceedings of the 29th International Symposium on Automotive Technology and Automation (ISATA)*, Florence Italy, 97-104.

van Dantzich, M., Robbins, D., Horvitz, E., and Czerwinski, M. (2002). Scope: Providing awareness of multiple notifications at a glance. In *Proceedings of the 6th International Working Conference on Advanced Visual Interfaces (AVI '02)*. Trento, Italy, ACM Press.

Vertegaal, R., Velichkovsky, B., and van der Veer, G. (1997). Catching the eye: Management of joint attention in cooperative work. *SIGCHI Bulletin 29*, 4, 1997.

Wahid, S., Allgood, C. F., Chewar, C. M., and McCrickard, D. S. (2004). 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),* Banff, Alberta, Canada, 167-172.

Weiser, M. and Brown, J. S. (1996). Designing calm technology. *PowerGrid Journal*, 1.01.

Whittaker, S., Terveen, L., and Nardi, B. A. (2000). Let's stop pushing the envelope and start addressing it: A reference task agenda for HCI. *Human-Computer Interaction*, *15*: 75–106.

Wickens, C. D. and Hollands, J. G. (2000). *Engineering Psychology and Human Performance*. Prentice Hall, Upper Saddle River, NJ, third edition, 2000.