



HAL
open science

Supporting Novice to Expert Transitions in User Interfaces

Andy Cockburn, Carl Gutwin, Joey Scarr, Sylvain Malacria

► **To cite this version:**

Andy Cockburn, Carl Gutwin, Joey Scarr, Sylvain Malacria. Supporting Novice to Expert Transitions in User Interfaces. ACM Computing Surveys, 2014. hal-02874746

HAL Id: hal-02874746

<https://inria.hal.science/hal-02874746>

Submitted on 19 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Supporting Novice to Expert Transitions in User Interfaces

ANDY COCKBURN, University of Canterbury

CARL GUTWIN, University of Saskatchewan

JOEY SCARR, University of Canterbury

SYLVAIN MALACRIA, University of Canterbury

Interface design guidelines encourage designers to provide high-performance mechanisms for expert users. However, research shows that many expert interface components are seldom used, and that there is a tendency for users to persistently fail to adopt faster methods for completing their work. This paper summarizes and organizes research relevant to supporting users in making successful transitions to expert levels of performance. First, we provide a brief introduction to the underlying human factors of skill acquisition relevant to interaction with computer systems. We then present our focus, which is a review of the state of the art in user interfaces that promote expertise development. The review of interface research is based around four domains of performance improvement: *intramodal improvement* that occurs as a factor of repetition and practice with a single method of interaction; *intermodal improvement* that occurs when users switch from one method to another that has a higher performance ceiling; *vocabulary extension*, in which the user broadens their knowledge of the range of functions available; and *task mapping*, which examines the ways in which users perform their tasks. The review emphasizes the relationship between interface techniques and the human factors that explain their relative success.

Categories and Subject Descriptors: **D.2.2 [Design Tools and Techniques]—User Interfaces; H.5.2 [User Interfaces]—Graphical User Interfaces (GUI)**

General Terms: Human Factors

Additional Key Words and Phrases: Expertise, novice to expert transition, shortcuts.

ACM Reference Format:

Andy Cockburn, Carl Gutwin, Joey Scarr and Sylvain Malacria, 2014. Supporting Novice to Expert Transitions in User Interfaces. *ACM Computing Surveys* X, X, Article XX (XXX 2014), X pages.
DOI: <http://dx.doi.org/10.1145/0000000.0000000>

1. INTRODUCTION

Graphical User Interfaces (GUIs) mediate most communication between humans and computing devices. Their success is partly due to their natural support for novice users – the phrase ‘see and point versus learn and remember’ [Shneiderman 1987] describes how novices benefit from being able to visually find salient interface elements and manipulate them through a metaphor of direct manipulation. However, the very characteristics that make GUIs effective for novices also cause them to fail in their goal of supporting experts, and GUIs often trap users into a ‘beginner mode’ of operation. The richness and power of human perception, cognition, and motor action is constrained by standard GUI mechanisms such as visual search and direct manipulation, which are easy to understand but which force the experienced user

Authors’ version

Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. 2014. Supporting Novice to Expert Transitions in User Interfaces. *ACM Comput. Surv.* 47, 2, Article 31 (January 2015), 36 pages.
DOI: <https://doi.org/10.1145/2659796>

into relatively slow and laborious action. Conversely, interfaces explicitly designed for experts (e.g., keyboard shortcuts or command-line interaction) allow high levels of performance, but only after extensive training. While the design of interfaces for novices *or* for experts has been well investigated, the design of interfaces that facilitate a transition from novice to expert performance is less well understood.

The problem of users failing to achieve expertise has been shown in many studies across a wide range of interactive tasks and contexts. In the area of Computer Aided Design, field studies demonstrated that long-term users often employ inefficient strategies for completing tasks [Bhavnani and John 2000], and log studies have shown surprisingly limited command vocabularies among experienced users [Matejka et al. 2009]. Similar findings have been shown for text editing [Rosson 1983], operating systems [Draper 1984; Doane et al. 1990], and spreadsheets [Nilsen et al. 1993]. Furthermore, studies of interface mechanisms designed to facilitate expertise, such as keyboard shortcuts, have shown that these mechanisms are seldom used [Lane et al. 2005; Alexander 2009].

Carroll and Rossen [1987] used the phrase ‘the paradox of the active user’ to encapsulate the tendency for users’ interface performance to reach an asymptote at a level of mediocrity. They explained this effect through two biases that people bring to their work: a *production bias*, which encourages reuse of known methods, in preference to finding better methods that may improve throughput in the long run; and an *assimilation bias*, in which people apply existing knowledge to interpret and solve new problems. Both biases lead users to continue with known methods of interaction, causing an associated tendency to miss opportunities for improvement. In scrutinizing this paradox, Fu and Gray [2004] additionally explained that preferred procedures for interaction are maintained over faster recommended alternatives when the preferred procedure is well-practiced and generally applicable, and when it provides fast, incremental feedback.

The tendency for users to asymptote at mediocre performance has serious and enduring implications for the productivity of millions of office workers. The tools used to conduct everyday office work, such as word processors and spreadsheet applications, have maintained substantially consistent point-and-click graphical user interface mechanisms across decades of interface revisions, and their expert methods of interaction, such as keyboard accelerators or ‘hotkeys’, have also remained relatively stable. Furthermore, recent interface releases have tended to reduce the visibility of expert methods for interaction – for example, many hotkeys available in Microsoft Word have no visual depiction at all (e.g., Ctrl-Alt-M for “New Comment”), and the extra effort required to discover the hotkey (e.g., a web search) may deter users from doing so.

The typical design of graphical user interfaces, therefore, allows users to quickly learn suboptimal methods for task completion (e.g., by pointing and clicking), which users tend to maintain for months, years, and decades of subsequent interaction. While the productivity loss associated with each command selection is relatively small, these losses are multiplied across dozens of selections of each command per day (e.g., each invocation of the ‘bold’ command), for dozens of distinct commands, for several applications, for hundreds of days each year, and for dozens of years. The total productivity loss is analogous to an office worker who never learns to touch type – each keystroke is a minimal cost, but days of productivity per year are lost due to inefficient interaction methods.

Although many studies have demonstrated a tendency for users to fail to attain expertise, until recently there has been relatively little research into interface methods that can promote and assist the transition to expertise (with some notable exceptions, reviewed in Sections 4-7). Consequently, while designers know the problem exists, interface design guidelines provide little assistance. For example,

Nielsen's [1993] influential usability heuristics give the abstract guidance 'Provide Shortcuts' (which research shows are unlikely to be used) and 'Help and Documentation' (often the user's source of last resort [Randall and Pedersen 1998]). Recently, however, there has been an increase in literature explicitly addressing the problem of how interfaces can be designed to promote and assist expert levels of performance. This paper provides a review and analysis of interface strategies that can promote expertise, as well as providing a foundation in underlying human factors that can explain their success and point to promising new directions.

Naturally, learning and skill acquisition have been the focus of substantial research in the psychology literature. Texts such as Anderson's [2005] general review of cognitive psychology dedicate multiple chapters to the topic, and more specific texts such as Schmidt and Lee [2011] scrutinize how humans develop motor skills. The problem for interface developers and researchers, however, is that while there is abundant literature on skill development, there is a lack of clarity on how the findings can be applied in user interfaces to assist the development of user expertise.

This paper provides a review of literature addressing the problem of how user interfaces can assist users in transitioning from novice to expert levels of performance. The review is presented in two parts. First, we present a brief summary of key relevant findings on human skill acquisition, predominantly from the psychology literature. Second, we describe our focus, which is research from Human-Computer Interaction on interface techniques that are designed to support skill development, and we analyze their relative successes and failures. The review of interface research is divided into four sections, each addressing a different domain of performance improvement, progressing from low level issues of interface control through to high level issues of task strategy. The four domains are as follows: *intramodal improvement*, which occurs within a single interface method, such as when learning to operate a new pointing device; *intermodal improvement*, which occurs across interface methods, such as between mouse selection and hotkey use; *vocabulary extension*, which concerns the breadth of the user's knowledge of interface features; and *task mapping*, which concerns the way in which users approach their tasks, including how they learn their tasks and the strategies they employ. Finally, we identify a research agenda for future work on supporting expertise development with user interfaces. The overall aims of the paper are to summarize and distill existing knowledge on expertise development with user interfaces, to highlight successful strategies that designers might consider for facilitating expertise development, and to motivate and direct further research on the topic.

2. HUMAN FACTORS OF LEARNING AND SKILL ACQUISITION

From an experimental-psychology perspective, learning and skill acquisition are often defined in terms of functions of memory [Schmidt and Lee 2011] – human memory is the repository for human experiences and understanding, so anything that has been learned is encoded in some form of memory. However, human memory is a vast research topic and a complete review is beyond the scope of this paper. Instead, we limit the review to summarizing the aspects of human memory that directly influence skill development with user interfaces – for example, spatial and proprioceptive memory strongly influence the user's ability to rapidly acquire interface targets, and consequently these functions are reviewed in Section 2.7.

Similarly, there is vast research literature on learning and education that is beyond the scope of this paper. As with memory functions, for brevity we limit coverage of this literature to summarizing key lessons that are applicable to skill acquisition with user interfaces. We refer readers seeking more general introductions to Anderson [2005] for general psychology, to Baddeley [1999] for a review of human memory, and to Thomas [2013] for an introduction to education.

This section focuses on underlying human factors of skill acquisition that are relevant to user interface design. It begins by describing three stages of skill development, and then reviews several factors influencing skill acquisition, including repetition, type and distribution of practice, the role of effort in efficient training, and how different forms of motivation and feedback affect training outcomes. Later sections explain how many of these human factors are employed in specific user interfaces that aim to improve users' transitions from novice to expert performance.

2.1 Stages of psychomotor skill development: cognitive, associative, autonomous

Various models of psychomotor skill acquisition have been proposed, with Fitts and Posner's [1967] three stage model being particularly influential. It describes how skills are developed through cognitive, associative, and autonomous phases, outlined in the following paragraphs. Learning to change gears on a manual car is a commonly-used example of this process, with novices at the cognitive phase forming initial models that might involve understanding the mechanics of a clutch and the need to disengage the motor from the gearbox, removing power from the motor while doing so, etc. During the associative phase the learner focuses on repeating the necessary actions to refine their appropriate synchronization. And once autonomous, the coordinated actions are performed as a single burst of unconscious activity, leaving the driver's cognitive capacity free for higher-level activities such as navigating or planning their day.

Cognitive phase

During the cognitive phase, initial conceptions of the activity are formed, predominantly learning *what* activities are to be done. Knowledge at this phase is substantially declarative and explicit, and might be communicated through verbal or written instructions, or through direct observation or visual images of the activity.

Performance of tasks at the cognitive phase is characterized by *controlled interaction*. Schneider and Shiffrin [1977] identified the following characteristics of controlled interaction: 1) it is slow; 2) it is attention-demanding, in that other similar tasks interfere with its execution; 3) it is serial in nature; 4) and it is strongly volitional, in that the activities can be avoided or immediately stopped. Task performance during the cognitive phase is also inconsistent, partially due to the learner's exploration of alternative strategies of what to do. Despite this variance, task performance improvements are largest during the cognitive phase.

The most effective training strategies during the cognitive phase focus on establishing the person's explicit conceptualization or understanding of the task. They therefore typically involve explicit instruction, provision of clear models, or some other form of feedback that assists with task conception. Psychology research examining effective training methods for each phase are reviewed in Section 2.8.

Associative phase

The associative phase (also referred to as the 'fixation' phase) is characterized by improvements in the motor actions used to execute the task. While the cognitive phase is dominated by attention to *what* is done, the associative phase is dominated by attention to *how* it is done. Performance improvements generally involve subtle adjustments of execution, resulting in smaller gains than the cognitive phase, but the performance variance decreases. The verbal, declarative and explicit understanding of the execution of the task that characterized the cognitive phase is largely unused during the associative phase.

The associative phase is enduring, with many tasks requiring years or decades of practice before transitioning from the associative phase to automaticity.

Autonomous phase

The autonomous phase represents the ultimate level of psychomotor learning, which is attained after prolonged and extensive practice. Key characteristics of *automaticity* are the opposite of those characterizing the controlled interaction at the cognitive phase. Schneider and Shiffrin [1977] identified the following characteristics of autonomous actions: 1) they are fast; 2) they are not attention demanding, in that other verbal or cognitive operations do not interfere with their execution; 3) they are parallel in nature, with various operations possibly occurring together; 4) they are not volitional, in that processing is often unavoidable. Experienced touch-typists are likely to have encountered the non-volitional component of automaticity – when interrupted, a few pending keystrokes will be typed as a burst of activity prior to dealing with the interruption. Furthermore, there is evidence that people are unable to voluntarily terminate execution of these bursts [Salthouse 1985].

When an individual's task performance moves towards automaticity there is a reduction in the mental and physical effort expended on the task [Kahneman 1973]. This reduction can be observed using measures such as pupil dilation (pupils enlarge with mental effort, e.g., [Hyönä et al. 1995]), galvanic skin response (skin conductivity increases with effort, e.g., [Engström et al. 2005]), or fMRI (e.g., [Hasegawa et al. 2002]). Dual attention tasks are commonly used to examine automaticity because the parallel processing and absence of attention that accompanies automaticity allows improved performance on concurrent activities, with relatively small detriment to the skilled execution.

2.2 The Power Law of Practice

The three-phase model of skill acquisition provides a useful characterization of how people conceptualize and enact tasks, as well as suggesting how different training interventions might assist skill development at each phase. Although the phases are not discrete and different individuals may progress at very different rates, when aggregate task performance is viewed across time, the performance curve follows a smooth and continuous power function, with substantial initial gains that gradually diminish. This aggregate 'power law of practice' has been reliably observed across many studies, including Snoddy's [1926] early studies of drawing mirror images, Crossman's [1959] study of factory workers rolling cigars, and Card et al.'s [1983] study of text-editing tasks. The generality of this practice effect was noted by Fitts [1964] and named the 'power law of practice' by Newell and Rosenbloom [1981], although Heathcote [2000] argues that an exponential law is more accurate.

2.3 Type of Practice

Although the power law of practice has been shown to apply in many domains, there is strong evidence that repetition alone is insufficient for the attainment of elite skill. Ericsson [2004] reviews multiple studies of elite performance across diverse domains (including musicians, athletes, and chess players), concluding that in addition to many thousands of hours of practice, the type of practice is critical. In particular, he demonstrates the necessity of *deliberate* practice, which focuses on tasks beyond the person's competence and comfort [Ericsson et al. 2007]. Deliberate practice facilitates two kinds of learning – improving skills already attained, and extending the reach and range of skills; it is further assisted through expert mediation, such as skilled coaching.

Ericsson refers to the cognitive, associative, and autonomous phases of learning (Section 2.1) when explaining the difference between the attainment of 'everyday skills' and 'expert' (or elite) performance. He argues that *everyday skills* are developed as described in Section 2.1, progressing through cognitive and associative phases, and becoming satisfactory once autonomous. However, he contends that elite

performance require deliberate practice to counteract automaticity during practice, with performers using various forms of mental representations to maintain the cognitive and associative phases [Ericsson 2004]. Further, he observes that once deliberate practice is abandoned, elite performance may deteriorate.

2.4 Depth of Processing and Effort

The deliberate use of mental mechanisms to assist continued learning and improvement described by Ericsson are reflected by the research of Craik and Lockhart [1972], who proposed a “levels of processing” framework in memory research. They postulated that the strength of a memory is a positive function of the depth to which the stimulus is analyzed. Several studies validated the framework – for example, Craik and Tulving [1975] showed that ‘deeper’ or more effortful mental manipulations during word memorization improved recall over ‘shallow’ encodings. In their experiment, word stimuli for memorization were coupled with shallow or deep questions, where ‘shallow’ encodings were based on word structure (e.g., “is the word in capital letters?”) or on phonetic properties (e.g., “does the word rhyme with ‘weight?’”), while ‘deep’ encodings were based on categories (e.g., “is the word a type of fish?”), or on sentence fitting (e.g., “would the word fit in the sentence ‘he met a _____ in the street?’”).

Regardless of the exact framework used to encapsulate memorization procedures, there is comprehensive empirical evidence that *elaborative processing* [Anderson 2005] (p.193), which increases the difficulty of processing stimuli for memorization, has a critical role in learning. This is an important and possibly counterintuitive finding for systems that are intended to improve ultimate user performance, because it suggests that interventions that cause temporary performance degradation during training may be beneficial in the long term – “manipulations that degrade the speed of acquisition can support the long term goals of training” (p.207, [Schmidt and Bjork 1992]). This issue is revisited in Section 2.9, which examines the *guidance hypothesis*.

2.5 Convergence and Divergence of Individual Differences with Practice

In some conditions, the variance between individuals’ performance decreases following practice, but in others, the variance increases. Studies on the role of deliberate practice and mental effort help explain why some people attain higher levels of performance than others despite their prolonged and intense repetition. As Ericsson puts it, “it may appear that excellence is simply the result of practicing daily for years or even decades. However, living in a cave does not make you a geologist” [Ericsson et al. 2007], p3.

Ackerman provides a succinct summary of the conditions under which individuals’ performance becomes more similar or more different as a result of practice [Ackerman 2007]. For simple tasks that involve speed and accuracy of motor movement, results suggest that differences diminish as a result of practice. For example, the individual differences between people first using a mouse to control a cursor might be large, but converges with practice. However, if a motor skill task is more complex or if it enables unobvious styles/strategies of use, then performance differences may stay constant or diverge with practice. For example, two novice keyboard users may begin with a ‘hunt-and-peck’ strategy of one-finger typing, but one user may dedicate deliberate practice to touch-typing, thus accelerating their performance, while the other does not.

As the touch-typing example suggests, as tasks become more complex or strategic, or as they demand more knowledge, there is a tendency for divergence to increase between the levels of highest and lowest performance.

2.6 Distribution of Practice

Altering the temporal distribution of training sessions can influence their effectiveness, and rest periods between sessions improve learning, with longer rest periods being more beneficial than short ones. These effects have been empirically demonstrated in the psychomotor domain for both short term activities (e.g., a point-following task with 30 second training batches and rests of between 0 s and 60 s, as used by Bourne and Archer [1956]) and long term activities (e.g., typing tasks trained once or twice daily for a total of 60 hours, as used by Baddeley and Longman [1978]). Similar effects have been demonstrated for cognitive learning. For example, Cepeda et al. [2006] present a review of 184 papers studying verbal recall tasks, demonstrating that spaced (versus continuous) practice sessions improve recall performance. Their review also reveals an interaction between the optimal spacing period and the intended duration of memory retention, with longer spacing periods working best for longer retention.

2.7 Spatial and Proprioceptive Memory

Spatial memory is particularly important for efficient interaction with graphical user interfaces. When an interface is spatially stable users can make rapid decisions about the location of target items, but when stability is compromised users must resort to comparatively slow visual search [Cockburn et al. 2007; Cockburn and Gutwin 2009].

Spatial memory is a substantial research field within psychology (Anderson [2005], chapter 4), but the findings reported above also apply to spatial memory. For example, the *method of loci* can be considered to be a form of elaborative processing that aids recollection by associating memory stimuli with known spatial locations (such as the rooms in a familiar house). Related ‘depth of processing’ spatial memory effects have been reported in studies such as Naveh-Benjamin [1987] and Van Asselen, Fritschy and Postma [2005], as well as several studies specifically concerning interaction with computer systems (reviewed later in Section 4.2).

The input to spatial memory may be represented by visual, aural or proprioceptive stimuli. While graphical user interfaces are predominantly portrayed visually, consistent proprioceptive actions for control is likely to yield significant advantages for expert users. For example, the consistent spatial location of a vehicle’s brake pedal (relative to the driver) facilitates rehearsal of precisely the same physical action to slow the vehicle, with obvious advantages in an emergency. Touch-typing allows equivalent performance benefits for acquiring consistently defined proprioceptive targets. Conversely, however, the need for different movements to a target dependent on varied initial cursor location may impair the ultimate performance ceiling and ability to attain autonomous levels of performance with mouse and cursor interaction (further explored later in Section 4.2).

2.8 Motivation and Feedback

Positive motivation to learn and improve is a key determinant in whether a learner chooses to participate in practice. Understanding how interventions influence motivation is important in areas such as industrial psychology, sports science, and interface design for expertise.

Two key strategies for improving motivation are to make the task seem important and to use goal setting [Schmidt and Lee 2011]. These strategies are not distinct – for example, a person may view performance improvement as unimportant until stimulated by the presentation of a goal representing the higher performance of others. Consequently, several studies have attempted to understand the types of goals that lead to the highest performance outcomes. Interfaces using related methods to promote skill development are reviewed in Section 5.1.

There is extensive literature on the effectiveness of different forms of feedback in assisting learning, particularly in the field of education. High-level guidelines are necessarily broad and abstract, such as ‘create a respectful, friendly, open-minded and unthreatening climate’, ‘base feedback on observed facts’, and ‘suggest ideas for improvement’ (e.g. [Hewson and Little 2001]). In general, studies agree that specific, absolute goals of moderate difficulty yield better performance and learning outcomes than non-specific goals (such as “do your best”) or no goals [Tubbs 1986; Kylo and Landers 1995; Locke and Latham 2006].

In the psychomotor domain, where feedback is applied to learning physical movement, there is potential to deploy psychology findings in user interfaces (where users execute motor actions to activate particular interface functions). Schmidt and Lee [2011], chapters 11 and 12, provide a comprehensive review, covering both the *intrinsic* feedback effects (such as visual or proprioceptive stimuli) that are inherently coupled with limb movement, as well as the effectiveness of different forms of *extrinsic* feedback that can be explicitly applied to assist skill acquisition. Many parameters of augmented feedback can be manipulated, including the following:

- *temporal properties*, such as providing feedback that is either concurrent with the action, presented on termination of the action, or delayed;
- *aggregation*, such as providing independent feedback about each *discrete* action versus *accumulating* information about a sequence of actions;
- *modality and form*, such as textual or spoken instruction, video of the action, or statistical summaries of performance;
- *knowledge of performance (KP)*, which concerns information about the way in which the action was executed, such as the deviation from the ideal movement. For example, a golfing tutor might show a video of a student’s swing, possibly commenting on foot position or backswing speed. Note that the feedback concerns the movement, not its outcome.
- *knowledge of results (KR)*, which concerns information about the outcome of the action that is presented after its termination, such as its success or failure, or the time taken to complete it. For example, a typing application might provide feedback in the form of a beep to indicate a target phrase or character was typed incorrectly, or it might show a plot of word-per-minute typing rate; similarly, a golfing tutor might note the distance that the ball travelled. Note that the feedback concerns the outcome, not the movement itself.

The distinction between *knowledge of performance (KP)* and *knowledge of results (KR)* has been extensively studied in the psychomotor literature. Gentile [1972] (who introduced the term ‘knowledge of performance’) postulated that KP is most effective for *closed* motor tasks, in which the task is uniformly performed without response to changing external factors, and where the movement itself is the goal of the skill – for example, a golf swing would be considered to be a closed motor task*. It is debatable whether certain forms of KP alone (such as non-augmented videos of performance) can yield substantial performance benefits – for example, Rothstein and Arnold [1976] showed that performance videos alone did not improve the participants’ performance, possibly due to a lack of specificity about the actions required to improve. However, it is clear that performance and learning outcomes are enhanced when KP is combined with KR (e.g., Wallace and Hagler’s study of basketball players [1979]). As with cognitive feedback, given the diversity of literature and their results, we defer details until reviewing interface methods in Section 4.2.

* Note that categorising tasks as ‘open’ or ‘closed’ is a different concept to open- and closed-loop motor control. The terminology is unfortunate because an open task requires closed-loop control.

2.9 Guidance

Physical guidance is a form of feedback where the learner's limbs are externally manipulated in an attempt to assist learning physical movements, and results vary on its efficacy. For example, Armstrong [1970] showed that for learning complex elbow movements, mechanical limb control resulted in precise movements during training, but inaccurate reproduction once the controls were released. In contrast, when training was achieved without mechanical control but with visual feedback of a motion plot (either at the end of the trial or concurrently during it), participants produced the motions most accurately (once feedback was removed) when trained in the post-movement feedback condition. In other words, delayed and unconstrained feedback achieved the best learning outcome. Not all experimental results agree with this finding, and it seems likely that some form of coarse physical guidance may assist learners in their initial conceptualization of the required movement, particularly if the movements are large and relatively slow (e.g., teaching a child a kayak stroke).

These results also emphasize important questions about the methodology used for evaluating skill acquisition and learning. In particular, the *guidance hypothesis* [Schmidt 1991] suggests that augmented feedback which improves early performance through guidance may impair retention of the performed skills once the guidance is removed. Consequently, most experiments in the psychomotor domain use separate experimental periods for *training*, *retention* (the trained tasks are evaluated later, often 24 and 48 hours after initial training), and *transfer* (where the skills are applied to a related but non-identical task).

Although rare in current user interfaces, results on physical guidance may be relevant to interfaces that use force- and tactile-feedback, such as training aids for dentistry procedures [Rhiemora et al. 2010] and teaching the blind to write signatures [Plimmer et al. 2008]. Furthermore, many user interfaces exploit some form of dynamic visual guide, or 'feedforward', to assist users in performing their task. We return to specific user interface examples in Section 4.

2.10 Other human phenomena influencing skill acquisition

Although many of the effects described above are empirically reliable, there are also various human phenomena that complicate the deployment and analysis of interventions intended to aid transitions to expertise.

Satisficing

Rather than seeking to continually optimize performance, people have a tendency to 'make do' with solution strategies that were first learned, even though they may be known to be suboptimal. Furthermore, once a suboptimal strategy has been learned and reproduced several times, it is likely to become habit, further reducing the likelihood of changing to faster alternative methods. Simon [1959] used the term 'satisficing' to describe how decision-makers often lack the information and cognitive resources needed to make a rational optimal decision, and also that once the costs of calculating a rational near-optimal decision are accounted for there will be little difference between attempts to optimize and making a faster, approximate 'satisfied' determination. In user interface research, satisficing has been used to explain phenomena such as users failing to make more extensive use of keyboard shortcuts despite their using substantially the same interface for years or decades (Section 5.1).

Gray and colleagues have extensively investigated the seeming paradox of prolonged suboptimal performance. Fu and Gray [2004] describe the tendency for users to persist with interaction methods that are well-practiced, are generally applicable, and which provide incremental feedback. Gray et al. [2006] also present evidence supporting their 'soft constraints hypothesis', which explains how short-term interactions (of duration between ½ and 3 seconds) can be construed as locally

optimal within those timeframes, although globally suboptimal. In explaining this seeming contradiction, they use the analogy of a person following instructions to build a toy – the globally optimal solution might involve memorizing the full set of instructions before assembling any components, but a locally optimal one might intersperse reading instructions with assembly.

Arousal and personality

The efficacy of certain training interventions has been demonstrated to interact with individual and environmental factors, which greatly complicates the process of determining which training methods are likely to be most effective for different people in different settings. For example, while it is known that performance generally follows an inverted ‘U’ shape with level of arousal (i.e., performance deteriorates when a person is under- or over-excited), there is evidence that personality differences such as introversion and extroversion result in different baselines of arousal, and consequently the same stimulus effect may increase performance for an extrovert while decreasing performance for an introvert [Revelle et al. 1980; Bullock and Gilliland 1993]. Revelle et al.’s study also showed interactions between several other factors, including time of day, other personality factors, and consumption of stimulants.

Interaction with skill complexity

Several studies have shown important interactions between the effectiveness of training interventions, the participants’ skill level, and the complexity of the skill conducted. Perkins-Ceccato, Passmore and Lee [2003] showed that low-skill golfers benefited more from instruction that focused attention on internal motor aspects of actions when compared to instructions that focused on external effects on the ball; conversely, highly-skilled golfers benefited more from externally focused instructions. Similarly, Wulf and Shea [2002] found that principles derived from the study of simple tasks do not generalize to more complex skills. They explained this effect with reference to the learner’s total cognitive load – when task demands are low, learners benefit from practice conditions that increase total load, whereas when task demands are high, practice conditions with more manageable workload are beneficial.

2.11 Summary of Human Factors of Skill Acquisition

Highly skilled task performance is characterized by automaticity, where the physical actions required for the task are executed quickly, efficiently, and with minimal conscious deliberation. Importantly, automaticity allows the performer to think about higher-level tasks during the execution of the skill. While high degrees of repetition are required for skilled execution to approach automaticity, different training interventions influence the way performers gain skill and understanding.

This section reviewed some of the key human factors influencing psychomotor skill acquisition, with a focus on those most relevant to interaction with computer systems. As stated earlier, the aim was to provide a brief introduction to assist in understanding the objectives and methods of the interface techniques described in upcoming sections.

3. FOUR DOMAINS OF INTERFACE PERFORMANCE IMPROVEMENT

The following sections review research examining user interface methods that assist users in transitioning from novice to expert levels of performance. Typically, ‘high performance’ will be exhibited by rapid task completion, but in some interactive contexts the focus of improving performance may be on error reduction, or on increasing the quality of the product. Our focus is on methods that can be deployed *within* the user interface, rather than the broader range of methods that are external

to the interface, such as offline training courses, the role of socialization in interface learning, or individual preference for learning style.

The review is structured by considering four domains that characterize different opportunities for improving performance with user interfaces. These domains are derived from a conceptual deconstruction of interaction with an interface as consisting of the user's *performance characteristics* with a set of *alternative methods* for activating a set of *functions* that are required to complete a *task*.

Functions are the set of interface commands and capabilities that allow data, state, or view manipulations to be achieved. For example, a simple painting application may support functions for drawing lines, painting shapes, zooming, scrolling, and so on. Analyses of functionally rich interfaces have shown that even experienced users typically use only a small subset of available functions [Draper 1984; Matejka et al. 2009].

Each interface function can typically be accessed through more than one interface *method*. For example, a line-drawing function might be selected by clicking a palette icon, through pull-down and context menus, or via a keyboard shortcut; and a scrolling function might be controlled by dragging a scroll thumb, by rotating a scrollwheel, or by pressing keyboard arrow keys.

Each *method* has associated *performance characteristics*, including a performance *floor* and *ceiling*, and each user will have attained performance ability somewhere between the floor and ceiling.

The four domains of interface performance improvement described in the following subsections are as follows:

1. *intramodal improvement* concerns the rapidity and magnitude of performance improvement with one particular interactive method (e.g., pointing with the mouse) for one particular function (e.g., selecting the bold function in a word processor);
2. *intermodal improvement* concerns ways to assist users in switching to faster methods for accessing a particular function (e.g., switching from cursor-based interaction to keyboard shortcuts);
3. *vocabulary extension* considers ways to help users broaden their knowledge and their use of the range of functions available in an interface;
4. *task mapping* addresses higher-level issues of the strategies that users adopt when seeking to complete their tasks with a user interface. It concerns the coordination of functions to complete a task.

Figure 1 visually characterizes the key objectives of intramodal improvement, intermodal improvement, and vocabulary extension, as well as their relationship with interface methods and functions. The height of each column in Figure 1a depicts

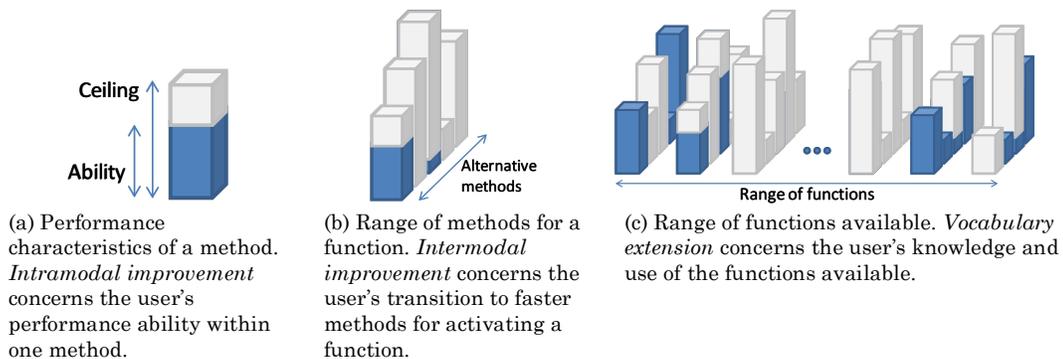


Figure 1. Conceptual representation of a user interface: **a.** methods that have distinct performance characteristics; **b.** alternative methods for any function; and **c.** a range of functions. *Task mapping* (not shown) concerns coordinating functions for a task.

the performance ceiling enabled by a particular interaction method for a particular function, and the blue shaded region depicts the performance level attained by the user (to reiterate, this characterization is not dependent on any particular measure of performance). Figure 1b depicts a range of alternative methods for one function, emphasizing that different methods have different performance characteristics. Figure 1c depicts the range of functions available in a user interface, showing that users may be unaware of certain functions. *Task mapping* (not shown in the figure) concerns coordinating a set of functions to achieve a task, and typically various combinations are possible for the same task.

4. DOMAIN 1: INTRAMODAL PERFORMANCE IMPROVEMENT

Based on Newell and Rosenbloom's [1981] power law of practice, Figure 2 depicts how user performance improves with experience of a *single* interaction modality. In the description below, we subdivide this curve into three segments for *initial performance*, *extended learnability*, and *ultimate performance*. These three stages are suggestive of Fitts and Posner's [1967; Anderson 1995] *cognitive*, *associative*, and *autonomous* stages of skill acquisition, described in Section 2.1.

The following subsections describe factors influencing each of these stages, including the observation that some factors assisting performance at one stage can impair performance at another.

4.1 Initial performance

Interface design for the initial stages of learning is strongly promoted in most usability guidelines [Norman 1983; Norman 1983; Shneiderman 1992; Dix et al. 1993; Nielsen 1993]. At this stage, users are unfamiliar with the interface, and must rely on their prior experiences, visual search, and recognition to find the commands they need. To optimize initial performance, designers aim to make interfaces easy to comprehend, with as high a performance floor as possible.

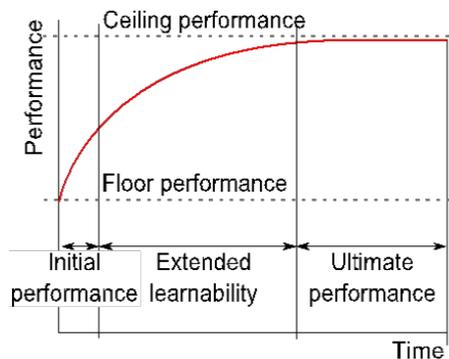


Figure 2. Characterisation of the intramodal power law of learning performance curve across time of use for one particular interface method for accessing one particular function.

Without prior experience to draw on, novice users are heavily reliant on visual search, and the time for visual search is a linear function of the number of candidate items that must be considered [Hornof and Kieras 1997; Wolfe 1998; Cockburn et al. 2007; Cockburn and Gutwin 2009]. Novice performance, therefore, can be improved by reducing the number of controls displayed to the user. Carroll and Carrithers [1984] exploited this effect with their 'Training Wheels' interfaces, which intentionally reduced the number of commands displayed. Related ideas were pursued with 'multi-layer interfaces' [Shneiderman 2003], which increase the level of functionality and reduce instructional interventions as the user becomes more experienced. However, by presenting different interfaces to novice and expert users,

there are risks of confusing users when making a transition to the more advanced user interface (introducing a ‘performance dip’, as discussed in Section 5).

‘Ephemeral Adaptation’ [Findlater et al. 2009], shown in Figure 3, demonstrates a different approach to helping users quickly identify important commands. With this

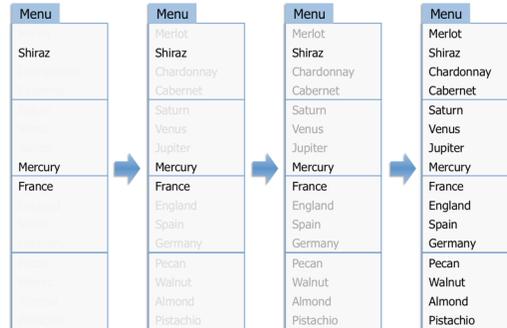


Figure 3. Ephemeral Adaptation. Reprinted with permission of ACM ©, from [Findlater et al. 2009].

The most likely target commands are shown immediately, while others are faded in over a few hundred milliseconds. This technique can be used to draw a novice’s attention to the most likely commands, while maintaining the original menu layout.

technique, a small subset of menu items are immediately shown when the menu is displayed, while the remaining items are gradually faded into view over a few hundred milliseconds.

The notion that controls should be visible to be learned is also well expressed in most usability guidelines, but the corollary of making novice functionality visible is that expert functions are often suppressed, reducing the likelihood that they will be discovered. A related concept is that appropriate interface controls should be ‘ready to hand’ [Karat et al. 2000]: that is, controls and feedback should be available for use but not obstruct task completion. Dyck et al. [2003] observe that many computer games achieve the dual objectives of availability without obstruction through ‘calm messaging’ that uses non-abrupt and non-intrusive means for presenting information, such as transient text, animation, and audio.

4.2 Extended learning

Several factors influence the rate at which performance increases after initial familiarization, including the effort required during training and the use of guidance to assist performance. These factors also apply during *intermodal* improvement.

Effortful learning

In seeking to exploit findings from psychology suggesting that “deeper” mental encodings result in stronger memories [Craik and Lockhart 1972; Schmidt and Bjork 1992] (Section 2.4), several interface researchers have examined interfaces that improve skill development. For example, Cockburn et al. [2007] examined the effects of using an intentionally effortful ‘frost brushing’ game to help users learn a gesture-based method for text entry, called ShapeWriter [Zhai and Kristensson 2003]. The game involved bursting ‘balloons’ containing target words – each word appeared at the bottom of the display and floated up and off the display unless burst by correctly entering the word (see Figure 4). When using the frost brushing interface, the visual guidance normally provided by the display of keys on the keyboard was occluded by ‘frost’ over the display. To show the underlying key, users had to brush away the frost by waving a stylus cursor over the key, and the frost quickly reformed. The design intention was to elevate deeper mental encodings of the spatial location of letters by enforcing the need for memorization. Experimental results showed that the frost brushing condition improved users’ memory of letter locations and of gesture

shapes compared to the traditional keyboard display (memory for gesture shapes was measured by asking users to create word gestures on an blank keyboard, with gesture quality reported by a recognition engine score function). However, the participants' comments revealed design subtleties in choosing an appropriate level of effort. In experiment on learning letter locations the frost brushing condition was rated as more engaging and less frustrating than the normal interface; but an experiment involving the harder task of learning gesture shapes for entire words showed the frost brushing interface to be less engaging and more frustrating than

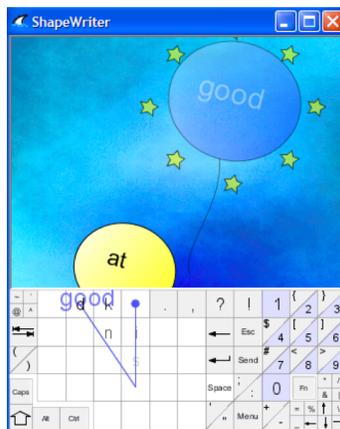


Figure 4. The 'frost brushing' training game for ShapeWriter [Zhai and Kristensson 2003], reprinted with permission of ACM ©, from [Cockburn et al. 2007]. Letters on the keyboard are occluded by 'frost' that quickly reforms after being 'scrubbed away' with the stylus. A balloon bursts when its word-gesture is correctly entered.

normal. The results suggest that there is a fine balance between engaging and frustrating users, and that training interfaces should be neither too mundane nor too difficult. This finding reflects those of Wulf and Shea [2002] who observed that principles derived from the study of simple tasks may not generalize to more complex skill learning (Section 2.10).

In examining interface methods that support the development of touch-typing skills, Yechiam et al. [2003] proposed using a secondary task during text entry that imposed a 'moderate and immediate punishment' for looking at the keys rather than the text output. While entering text, the secondary task required a rapid response to the periodic display of a blue square in the text output area. If the user failed to quickly respond to the blue square's appearance, then the entire display would be temporarily dimmed. Users were much more likely to fail to notice the blue square if looking at the keys rather than the display. Experimental results confirmed that the inclusion of the secondary task improved participants' touch-typing performance by encouraging users to rely on key memorization (and look at the text output area), rather than focus on the keyboard.

A recent study by Kim and Ritter [2013] investigated learning, forgetting, and re-learning a spreadsheet task when using mouse-and-menu interaction methods in comparison to keyboard-command interaction. Their findings suggest a tendency for mouse-and-menu users to forget methods sooner than the keyboard-command group. This finding is consistent with the depth of processing hypothesis, as suggested by Kim and Ritter's characterization of mouse-and-menu interaction as relying predominantly on 'knowledge-in-the-world' in contrast to keyboard-command interaction relying on 'knowledge-in-the-head'.

Revelation, guidance, and rehearsal in marking menus

Kurtenbach et al.'s research on marking menus makes several substantial contributions on supporting natural transitions from novice to expert levels of performance with user interfaces [Kurtenbach 1993; Kurtenbach and Buxton 1993; Kurtenbach et al. 1993; Kurtenbach and Buxton 1994; Kurtenbach et al. 1994].

Figure 5, extracted from Kurtenbach et al. [1994], shows both the novice (left) and expert (right) modes for hierarchical menu item selection when using a marking menu. Marking menus are a form of radial menu, in which directional movements from a starting location select different items. Shortly after the user's stylus makes contact with the surface (Figure 5, left) a radial menu is displayed; the user moves northward to select the 'Groceries' submenu, causing a sub-menu to be displayed; and moving rightwards then selects the 'Fruit & Veg' item. Alternatively, an expert user (shown in Figure 5, right) can make a rapid gesture of similar shape to quickly select the same item, but without needing to pause for guiding feedback.

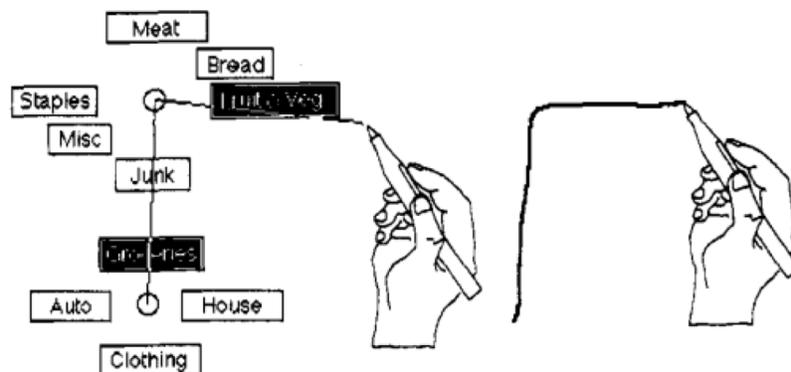


Figure 5. A hierarchical marking menu selection being made in response to visual guidance (left), and without visual feedback using a previously rehearsed action (right). Reprinted with permission of ACM ©, from [Kurtenbach et al. 1994].

Marking menus were designed to adhere to a set of design guidelines, including *revelation, guidance, and rehearsal* [Kurtenbach et al. 1994], as follows:

Revelation is used to provide interactive feedback to the user about the commands that are available and how the user can invoke them. Revelation is particularly important for gestural interaction, where the user's actions need not have any associated representation on the display. Unlike buttons, scrollbars, and other graphical widgets that afford a particular method for activation (clicking, dragging, etc.), gestural interactions need not be conducted with reference to specific graphical entities. The *revelation* mechanism used by Kurtenbach et al.'s marking menus required a dwell timeout to expire before graphical feedback of the radial menu items was presented.

The *guidance* principle states that the methods used for revelation should assist the user in specifying the complete command. In particular, guidance should not interfere with the user's specification of intention. In the context of a hierarchical marking menu, the appearance of a second-tier radial menu part way through a gestural command (such as 'Meat' to 'Staples' in Figure 5) serves as *guidance*, while a pop-up dialogue box that removes attention from the task would not.

Rehearsal states that the way that guidance is provided should require a physical rehearsal of the expert's actions. In other words, the novice's motor actions should rehearse the expert's motor actions to assist the development of automaticity and muscle memory.

Guidance versus effort for attaining performance versus learning

Kurtenbach et al.'s marking menu research was conducted in the early 1990s, with a focus on stylus-based input. The recent surge in popularity of finger-based gestural interaction on mobile devices has renewed interest in advanced techniques for easily learned and high performance gestural interaction techniques.

'OctoPocus' [Bau and Mackay 2008], shown in Figure 6, uses dynamically adapted guidance to assist users in performing and learning gestures. The technique assists performance of specific gestures (intramodal improvement) as well as learning other gestures that have similar initial gestural shapes (vocabulary extension). When the finger makes contact with the surface, the available gestures are shown, with initial paths emphasized and fading with distance. Once the user begins tracing the gesture, the 'feedforward' portrayal of the required remaining gestural action adapts to the completed gesture components – for example, the path for 'Paste' is removed in the right-hand image of Figure 6, and the path for 'Cut' is deemphasized by making its path thinner.

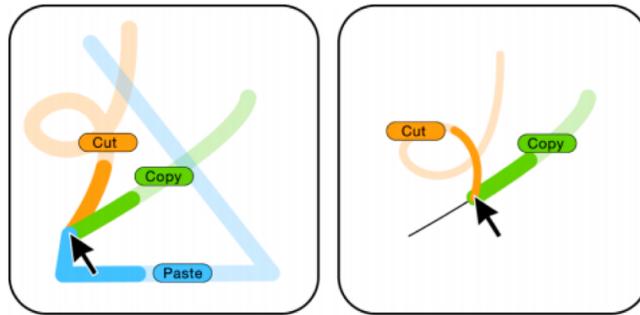


Figure 6. Dynamic gestural guidance with OctoPocus. Reprinted with permission of ACM ©, from [Bau and Mackay 2008]. Initially (left), three alternative gesture commands are possible, but as the user gestures upwards and rightwards, guidance for 'Paste' is removed and 'Cut' fades, leaving only the remaining guidance for 'Copy'.

related gesture prediction methods: [Freeman et al. 2009; Appert and Bau 2010; Bennett et al. 2011; Kristensson and Denby 2011].

Anderson and Bischof [2013] conducted experiments to determine whether Schmidt's [1991] *guidance hypothesis* applies with guidance-based interactive methods (Section 2.9). They compared user performance using four different interfaces for providing guidance: 'crib notes', which provided static representations of the gestures, continually displayed in the corner of the display; 'static-tracing' representations, which showed non-responsive versions of all gestures at the starting location of the gesture; 'dynamic-tracing', equivalent to OctoPocus; and 'adaptive-tracing', which was identical to the 'static-tracing' condition except that the guide disappeared at a progressively earlier point through the training set (disappearing at the end of the gesture in the first training event, and at the start of the gesture in the last training event). Importantly, their experimental method analyzed both performance during training, as well as learned retention of the gestures, measured 24 hours after the training session.

Their results confirmed the guidance hypothesis – the conditions that maximized performance during training (static- and dynamic-tracing conditions) performed the worst in the retention tests, and vice versa. These results reflect those generated by Cockburn et al.'s [2007] frost-brushing experiments, in which the frost-brushing condition slowed user performance during training, but resulted in better results during subsequent testing conditions. Similar results have been generated for

intermodal improvement, such as Grossman et al.'s [2007] study of hotkey learning, described in Section 5.1.

These results have important implications for the design of training systems, as designers must understand whether their goal is to assist users in rapidly attaining high performance, or whether learnt outcomes and the ability to transfer the learning to related interactions are primary objectives.

4.3 Ultimate performance

The final characteristic of the intramodal curve is the asymptote, or performance ceiling (Figure 2). Card, Moran and Newell [1983] provide models and empirical evidence of 'expert performance of routine tasks', including analysis of one user who repeated the same editing task thousands of times to study the progression to automaticity. Scarr et al. [2011] identify four interface characteristics that contribute to high performance ceilings, as follows.

Flat command structures

GUIs typically contain more controls than can be easily displayed at once, necessitating interface partitions such as windows, tabs, and menu hierarchies. Navigating through these partitions takes time, and consequently there are potential performance benefits in flattening the command structure to make more items accessible at once [Scarr et al. 2012]. Commands issued by command line interfaces (CLIs) and hotkeys are exemplars as they have global interface scope (e.g. <Ctrl>-C executes 'copy' regardless of the interface state). Several research and commercial systems have used CLIs to improve interface performance: e.g., Quicksilver[†], Spotlight[‡], Enso[§], and GEKA [Hendy et al. 2010]. Although empirical results for CLI benefits over GUIs have been mixed (e.g. Whiteside, Jones, Levy and Wixon [1985]), it is widely accepted that CLIs enable higher efficiency, and power users are strong advocates (e.g. [Barrett et al. 2004]).

Terse and expressive input

Powerful interface commands communicate a lot of meaning in rapidly expressed actions. For example, a single alphabetic character can discriminate 26 commands, or 52 with case sensitivity; increasing to 2704 with two case-sensitive characters. However, there is often a tension between supporting terse, expressive power and meaningful mappings: for example, Alt-shortcuts in Office 2007 allow access to most interface controls, but they are often abstract and hard to remember (e.g., the key sequence '<Alt> n, nu, t' inserts a page number in Microsoft Word).

Revisitation/history support

Users' interactive behaviour is often repetitive (e.g., command use [Greenberg and Witten 1993] and web navigation [Tauscher and Greenberg 1997]), and interfaces can aid efficiency by explicitly supporting command repetition. For example, web browser URL address bars and the Google search box memorize previous activities and offer type-ahead shortcuts for them: e.g., the keystrokes 'cn<Return>' become a shortcut for a user who frequently visits CNN's website. Like marking menus, such interactions lie on the cusp with intermodal transitions, but we categorize them within intramodal improvement because the initial mechanisms for interaction are identical for both novice and expert modalities.

[†] <http://www.blacktree.com>

[‡] <http://support.apple.com/kb/HT2531>

[§] <http://humanized.com>

Spatial predictability

Studies have demonstrated that spatial stability allows users to make rapid decisions about items' locations rather than relying on comparatively slow visual search (e.g. [Cockburn et al. 2007]). Despite the desirability of spatial stability, this principle is often compromised due to display space constraints or to reconfiguration of the display – interface controls are often elided and repositioned as window geometry is manipulated, and this is necessary because widgets typically do not scale. However, this is a technical, rather than a human, limitation: Scarr et al. [2013] examined the robustness of users' spatial memory to various forms of spatial transformation (including scaling), finding that users were able to quickly locate items in spatially stable views that had undergone a uniform spatial transformation.

In their 'InfoCockpit' project, Tan et al. [2001] examined whether items displayed on multiple monitors wrapping around the user would enhance interaction based on the spatial predictability of items. Spatial audio was included to assist memorization of place, and their results showed a 56% increase in object memory compared to standard desktop systems.

Section 8 provides a brief summary of key findings for each of the domains, including intramodal improvement.

5. DOMAIN 2: INTERMODAL PERFORMANCE IMPROVEMENT

Most interfaces support more than one interaction method for accessing the same function (Figure 2b). Once a user approaches their intramodal performance asymptote, any further performance improvement requires shifting from one method to another that offers a higher performance ceiling. For example, a user's performance in clicking a word processor's bold button will improve as they learn the icon's location, but substantial further performance improvement is possible if the user switches to a keyboard shortcut instead. Intermodal performance improvement concerns these transitions between interface methods.

However, Scarr et al.'s [2011] analysis of *intermodal* performance improvement postulated the existence of an important 'performance dip' that occurs when the user switches to another method, as depicted in Figure 7. The existence of this performance dip is likely to deter users from switching to new methods for interaction, even though doing so may offer long term productivity gains – the prospect of encountering a short term productivity loss may cause users to postpone switching to the new method, possibly forever. Factors acting as deterrents or

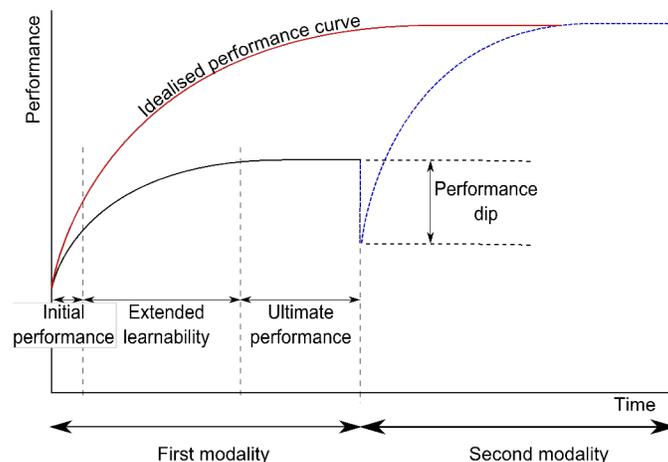


Figure 7. Performance curves characterising improvement within one interface method (left, 'intramodal improvement') and the performance dip that occurs when switching to a new, higher performance ceiling, method (right, 'intermodal improvement'). Adapted with permission of ACM, from Scarr et al. [2011].

barriers to intermodal transitions, and methods to overcome them, are reviewed in the following subsections, which address two critical points on the *intermodal* performance curve shown in Figure 7: first, factors influencing the initial switch to a new interface modality; and second, the performance dip that a user is likely to experience when switching from a familiar interface to an unfamiliar one.

5.1 Making an initial switch

Awareness of the new modality

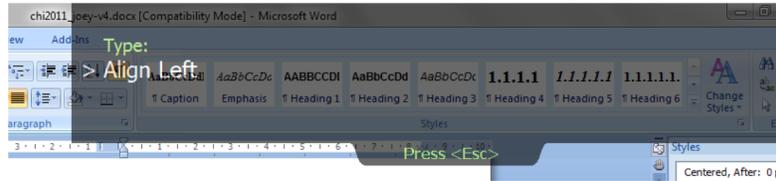
The first stage of supporting a transition to an alternative method for interaction is to make the user aware of the modality. Such awareness can be achieved through a wide variety of means, with varying impact and effectiveness. For example, traditional menus typically display a keyboard shortcut for each menu item, but users may not attend to them; and toolbar items often display rollover tooltips that include the hotkey, although a dwell timeout prior to displaying the tooltip means that users are unlikely to see the hotkey during normal interaction.

Interfaces can be forceful in their awareness mechanisms, requiring users to experience the new modality by demanding that actions are completed using it. Grossman [2007] experimented with a variety of schemes for assisting hotkey learning. These included visual and audio schemes to expose users to the hotkeys, a delay-based technique to deter use of the GUI (i.e., making the system unresponsive for 2 seconds after each selection), and a technique that forced hotkey use after each menu selection. Their results showed that forced use and audio feedback worked well, with 72.8% and 66.6% of experimental selections completed with hotkeys, and no adverse subjective response. In similar work, the HotKeyCoach [Krisler and Alterman 2008] showed a dialog box whenever an item was selected with the mouse, requiring either an extra click to proceed, or the hotkey to be issued. Although experiments showed these methods to be successful in increasing hotkey use, forced use of hotkeys and audio feedback may be unacceptable to users outside lab settings.

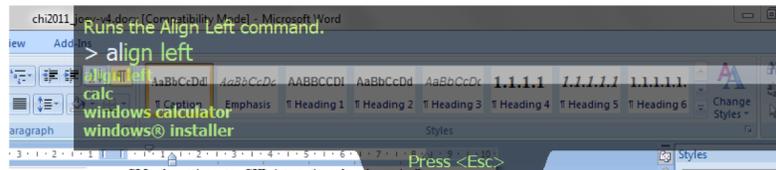
A less forceful approach to awareness was demonstrated by a system called 'Blur', which aimed to promote a transition from novice point-and-click interaction to higher performance command-line interaction by supporting command-line control of unaltered commercial GUI interfaces [Scarr et al. 2011]. When a GUI command was selected with the mouse, Blur's 'calm notification' showed a transparent window that temporarily slid onto the corner of the screen, displaying a command line alternative for the selection – the aim was to make command information available, but without requiring or demanding attention. Figure 8a shows Blur's calm notification window overlaid on the Microsoft Word ribbon following selection of the Align Left toolbar icon () . To use Blur's command-line, the user pressed the Escape key, typed part of the command to show alternatives, and pressed return to select. Figure 8b shows Blur's command recommendations after the user types '<Esc> al'.

Perception of the new modality

Once the user is aware of an alternative modality, the probability that they will switch to using it is influenced by their perception of its future efficiency, so all of the intramodal factors described above play a role. Importantly, though, several studies have demonstrated that predicted and actual experience differ (e.g. [Czerwinski et al. 2001]), and that users can mistrust their abilities, leading to under-estimates of potential benefit with the new modality. For example, Cockburn and McKenzie [2002] showed that users predicted that they would perform poorly in a spatial task, but subsequently rated their actual performance much higher. Similarly, studies have consistently shown that keyboard shortcuts offer a higher performance ceiling than mouse selection [Odell et al. 2004; Lane et al. 2005; Malacria et al. 2013], yet



- (a) Blur's calm notification of the command line alternative ('Align Left') for the GUI selection of the  toolbar item. The window disappears after 1 second and can be clicked through so as to not prevent pointer-based interaction with underlying widgets.



- (b) Command recommendation and command line completion in Blur. The user has typed 'al', and the first predicted command is 'align left', which is selected by pressing Return.

Figure 8. Blur's transient and transparent user interface. Reprinted with permission of ACM ©, from [Scarr et al. 2011].

Tak et al. [2013] found that some participants did not use known hotkeys because they believed toolbar selections were faster.

Even when users correctly predict the benefits of switching to a new modality, they may not do so due to the 'paradox of the active user' in which users "are likely to stick with the procedures they already know, regardless of their efficacy" [Carroll and Rossen 1987]. In seeking to understand this paradox, Fu and Gray [2004] formed theoretical cognitive models of different tasks and gathered data traces of their actual execution. Their findings showed that users are biased towards procedures that are (1) well-practiced and generic, and (2) composed of interactive components that are fast and incremental. They noted that these biases tend to impair task completion times.

To encourage users to improve their efficiency, Malacria et al. [2013] proposed the use of 'skillometers', which are lightweight interactive components that encourage users to reflect about their performance during interaction. Figure 9 shows an example skillometer evaluated in [Malacria et al. 2013], which shows the time taken to complete recent interface tasks, together with the time that might have been taken if an alternative method were used. The area labelled A in Figure 9 shows the time taken to select each of the last six commands (red if selected by pointing with the mouse; green if selected using a hotkey), as well as showing the hotkey binding for each of those commands. The area labelled B shows an aggregate performance meter, which is programmed to weight the most recent selection most highly to provide a clear 'reward' for hotkey use; and area C shows an estimate of the total time that might have been saved by switching to hotkeys. Comparative evaluation of a simple task with and without the skillometer showed that it caused a substantial increase in hotkey use. The 'Search Dashboard' [Bateman et al. 2012] is another example of a reflective widget (a form of skillometer), that improved user performance in web search queries.

Malacria et al.'s skillometer calculated time savings based on Keystroke Level Model [Card et al. 1983]. However, there are interesting possibilities in exaggerating the benefits to overcome the inertia induced by satisficing. Adar et al. [2013] provide a general discussion of the role of 'benevolent deception' in user interface design; however, if the deception is obvious users will distrust the system and stop using it.

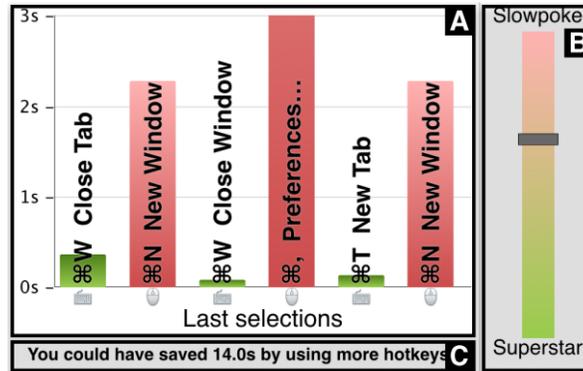


Figure 9. A skillometer widget, designed to encourage users to reflect on their own performance with a user interface and assist them in transitioning to interaction methods that offer a higher performance ceiling. Reprinted with permission of ACM ©, from [Malacria et al. 2013].

5.2 Performance dip after switching

The size of the performance dip that occurs after switching to a new interface will be influenced by the magnitude of the differences between the interface techniques (methods, functions, or strategies) used before and after switching. The *rehearsal* methods used by marking menus (Section 4.2) demonstrate minimising the performance dip (or eliminating it) by making the novice interface technique a rehearsal of the expert one.

‘ExposeHotKey’ [Malacria et al. 2013] used Kurtenbach’s [1993] *rehearsal* design guideline to promote hotkey learning and use. Normally, keyboard shortcuts are only displayed when the pointer rolls over a toolbar item or when a menu is shown. However, once the pointer has reached the item or menu, pointer-based selection is almost complete, so the user is likely to complete selection by pointing. Consequently, the user’s *intention* to learn a hotkey must be mediated by a non-hotkey modality (the pointer), which violates the principle of rehearsal. To facilitate hotkey rehearsal, ‘ExposeHotKey’ displayed all keyboard shortcuts at once, overlaid on top of the normal graphical widgets, whenever a modifier key is pressed. Consequently, users could *reveal* and *rehearse* hotkeys without using the pointer. Experimental results showed that ExposeHotKey led to substantial increases in hotkey use, with 94% of selections completed using hotkeys with ExposeHotKey, compared to 50% without it.

6. DOMAIN 3: VOCABULARY EXTENSION

Many interfaces allow access to extensive functionality, often including hundreds or thousands of commands [Hsi and Potts 2000; McGrenere et al. 2007], yet users typically know only a small subset. For example, a study of expert Unix users showed that they typically used ~45 of the ~570 commands available [Draper 1984]; and recent analyses of AutoCAD have shown that command vocabularies are typically ~30-40 among the ~1000 available [Matejka et al. 2009; Li et al. 2011]. This implies that users’ performance is often impaired by not using appropriate functions, possibly due to a lack of awareness or to subconscious satisficing.

The conceptual deterrents and barriers to vocabulary extension are similar to those described for *intermodal improvement* – users need to be aware of commands before using them, and they need to understand the magnitude of improvement that they will enable. Consequently, several of the methods described in the *intermodal improvement* section could also be used to increase the user’s vocabulary, such as calm notification or skillometers. Similarly, the methods used for explicit task instruction (described in Section 7), could also be used to extend the user’s vocabulary. In this section, we focus on other interface methods for extending

vocabulary, particularly those that use ambient suggestions and recommendations, rather than explicit instruction.

Researchers at Autodesk have conducted several studies aimed at improving users' knowledge and use of helpful interface functions. Two interrelated themes of their research concern understanding the best interface mechanisms for *presenting* recommendations to users, and devising methods to *generate* high quality recommendations. Their key interface goals are that the assistive content should be dynamically updated to maintain contextually relevant assistance [Ekstrand et al. 2011]; that it should be continually available to the user; and that it should be presented in an ambient manner that allows access at a glance, thus minimizing impediments to access [Matejka et al. 2011]. Figure 10 from Matejka, Grossman and Fitzmaurice [2013] shows their 'Patina' interface adaptation method, which uses colored heatmap overlays to emphasise command recommendations without impeding access to them. Related concepts of using stencils to direct the user's attention to salient interface objects were also presented by Kelleher and Pausch [2005] (in online tutorials).

In generating high quality recommendations, they focus on algorithms that produce recommendations that are both novel (rather than familiar) and useful [Matejka et al. 2009; Li et al. 2011]. They compared two forms of collaborative filtering algorithms: user-based algorithms, which generate recommendations based on the group of individuals that a particular user is most similar to, selecting those commands that are frequently used by the group of similar users; and item-based algorithms, which generate recommendations based on the similarity between the user's active command and other commands selected by the community of users. Evaluation results of their 'CommunityCommands' system suggest that an item-based collaborative filtering algorithm performs best, generating the highest number of useful recommendations as well as the fewest number of unuseful ones. While CommunityCommands uses one form of collaborative information to assist vocabulary extension, several other researchers have noted that other forms of social influence are also important. For example, Banovic et al. [2012] showed that social influence was important for encouraging tool palette customization; Peres et al. [2004] found that users were more likely to use keyboard shortcuts if they worked with others who did so; and Bateman et al. [2012] found that the queries issued to a search engine became more expert when the interface showed aggregate information about the user's past searches together with examples from experts' profiles.



Figure 10. Patina uses a coloured heatmap to recommend commands to users. Reprinted with permission of ACM ©, from [Matejka et al. 2013].

7. DOMAIN 4: TASK MAPPING AT THE INTERFACE

Domains 1-3 (intramodal improvement, intermodal improvement, and vocabulary extension) stem from a structural decomposition of the user interface into the performance characteristics of methods and the various methods and functions supported by the user interface (Figure 1). Domain 4, in contrast, focuses on interface techniques that assist with task comprehension, strategy selection, and customization.

7.1. Task training: Manuals, Online Help and Tutorials

A new user of a computer system must find ways to understand the system and the functions and methods that can be used to execute their tasks. In early HCI research, several studies observed the successes of exploratory learning in video games, leading to suggestions that designers of office systems should seek ways to intrinsically motivate users to explore the interface [Malone 1980; Carroll 1987; Shneiderman 1987]. However, subsequent studies have shown that exploratory learning for intrinsic interest is rare. Instead, users tend to focus on accomplishing real work tasks, and resort to tutorials, manuals, and trial-and-error as ‘just-in-time’ coping strategies [Rieman 1996]. Interfaces to online help systems have evolved substantially since Rieman’s study, but more recent studies indicate that online help is still predominantly used as material of last resort [Novick and Ward 2006]. However, this situation may evolve as systems improve at extracting the user’s task context (e.g., [Ekstrand et al. 2011]) and adopt ambient presentation to assure that help is continually ready to hand (e.g., [Matejka et al. 2011]).

Like online help, there has been a recent surge of research on online tutorials, with contributions from Autodesk being particularly prominent. Four emerging directions are described in the following paragraphs: gamification, community input into tutorials, leveraging transfer effects, and distribution of practice.

Gamification

Deterding et al. [2011] define gamification as “the use of design elements which are characteristic for games in non-game contexts”. Typically, gamification is used to increase the user’s motivation to engage in an activity and to improve their performance at it, although other benefits such as increased collaboration and creative leadership may also occur [Reeves and Read 2009]. McGonigal [2011] describes four key characteristics of games that engender intrinsic motivation, reflecting those described in Section 2.8: (1) provide a goal to give the user a sense of purpose; (2) use rules that limit how the goal can be achieved; (3) provide feedback to show progression towards the goal; and (4) voluntary participation, which engenders acceptance of the rules and feedback.

Researchers have contemplated or demonstrated gamification in a wide range of application domains, including education [Linehan et al. 2011], in-car aids [Diewald et al. 2013], learning software applications [Dong et al. 2012; Li et al. 2012], software calibration [Flatla et al. 2011] and business practice [Reeves and Read 2009]. As yet, there have been relatively few formal studies of gamification’s success. Two exceptions are Flatla et al. [2011] and Li et al. [2012]. Flatla et al. proposed using gamification to encourage users to undergo software calibration procedures that are sometimes necessary to optimize software to specific user characteristics, and their lab studies showed that users found gamified calibration procedures to be significantly more enjoyable, without compromising the calibration outcomes. Li et al.’s [2012] ‘GamiCAD’ tutorial system was designed to provide an engaging introduction to high functionality AutoCAD software. GamiCAD incorporated various gamified components, including a ‘mission console’ for navigating the game, a scoreboard, speed bonuses, step-by-step instructions, and various task-specific mini-games. Formal evaluation of GamiCAD in comparison with a non-gamified tutorial showed that participants completed tasks faster and more accurately when using GamiCAD, and that they preferred using it to a traditional tutorial.

Gamification is an emerging area of research, and we anticipate future results from field-testing studies.

Community input into tutorials

Lafreniere et al., [2013] describe and evaluate a tutorial system called ‘FollowUs’, which aims to integrate online tutorials into the associated application. They argue

that the lack of integration creates several limitations in tutorial material, particularly the inability for individuals and communities to update and improve tutorial content. FollowUs uses an ‘application-in-tutorial architecture’, where the application lies within the tutorial system, allowing the tutorial to capture the user’s workflow while following the tutorial, and providing easy mechanisms to record alternative or improved methods that might augment the original tutorial.

A formative evaluation compared user performance in a series of tasks when using FollowUs and when using a condition that differed only in the exclusion of community-sourced tutorials. Task completion rates were higher and frustration lower when using FollowUs.

In general, recent work on crowd sourcing and improvements in recommender algorithms are creating interesting opportunities for bringing relevant tutorial material to the user’s attention. However, Lafreniere et al. note that there are substantial challenges in designing interfaces that present this content to users in a manner that is helpful and timely, without being distracting.

Systems that exploit transfer effects between software versions

Ramesh, Hsu, Agrawala and Hartmann [2011] examined interface methods to help users transfer existing knowledge of one software system or version to another related system/version. Their ShowMeHow system allowed users to inspect the translation of commands between applications within the same domain, incorporating two main capabilities: the ability to click on a façade representing the known software to reveal related components in the new system; and text searches for command names in either application, based on manually constructed data structures describing terms used in both the original and new applications. An initial user study showed that users could locate unfamiliar commands in an interface much faster when using ShowMeHow, and a second study demonstrated that users could repurpose tutorials written for one application to learn another.

Using temporal distribution of practice in training systems

Finally, HCI researchers have used psychology findings on the effects of temporal distribution of practice (Section 2.6) to assist with learning motor aspects of interface performance. For example, studies by Zhai et al. [2002; Zhai and Kristensson 2003] examined users learning new keyboard layouts, with results suggesting effective memorization when users were trained using expanding rehearsal intervals.

7.2. Support for advanced task strategies

Bhavnani and John [2000] observe that knowledge of tasks and tools is insufficient for users to become efficient with user interfaces, and that users additionally need to attain an intermediate ‘strategic’ layer of knowledge. This strategic knowledge facilitates the efficient mapping from user tasks to system tools. Delaney et al.’s [1998] investigation of task performance also shows that overall task improvement is best analyzed with consideration to the user’s strategy.

Bhavnani and John’s analysis of strategic interaction highlights efficient strategies that generalize across a variety of application domains. For example, they contrast the novice strategy of ‘Sequence by Operation’ (exemplified by the window drawing task shown in Figure 11, top) with the more expert ‘Detail, Aggregate, Manipulate’ strategy (shown in Figure 11, bottom); other strategic examples are ‘Aggregate-Modify (all)-Modify (exception)’ and ‘Locate-Aggregate-Manipulate-Modify’. Bhavnani and John provide strong examples of the expertise benefits enabled by supporting these strategies in word processors (e.g., through the use of formatting styles) and Spreadsheets (e.g., formulae that work across multiple cells).

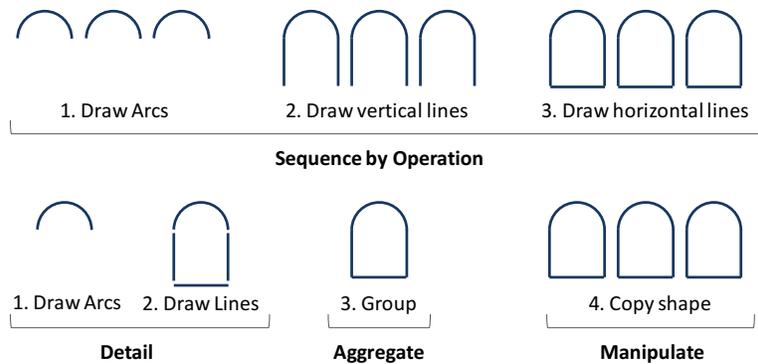


Figure 11. Two strategies for drawing three windows. Novice’s “sequence by operation” strategy (top); more efficient “detail-aggregate-manipulate” strategy (bottom). Adapted with permission of ACM, from [Bhavnani and John 1996].

More recently, researchers have examined the use of automated capture, analysis and review of workflows [Grossman et al. 2010; Kong et al. 2012], to assist users in reflecting on their interaction strategies. For example, the Chronicle system [Grossman et al. 2010] provides a zoomable timeline that allows the user to visualize and examine workflows for completing image editing tasks, with formative evaluations suggesting that they are useful tools. Autodesk now releases Chronicle as a technology preview, so results of field-testing may appear in future research.

7.3. Task disruption and locus of control

Feedback relevance and likelihood of disrupting the user are important considerations for any interface. Research by Bødker [1995] suggests that there are two types of task interruption: *breakdowns* and *focus-shifts*. Breakdowns result in severe disruption, forcing the user’s attention to a new activity. Focus-shifts cause only a brief attention switch, and cause less disruption. Interfaces such as Microsoft’s Clippy have been shown to be quickly abandoned after a few untimely interruptions [Shroyer 2000]; but conversely, less obtrusive feedback, such as Blur’s ‘calm notification’ [Scarr et al. 2011], may go unnoticed, resulting in a missed opportunity for improvement.

The related concept of locus of control concerns the nature of the stimulus for presenting information to the user. Information may be *pushed* by the system without any apparent triggering action from the user – for example, a ‘tip of the day’ message [Fischer 2001]. Alternatively, the user may explicitly pull the information – for example, intentionally dwelling on a button to view a hotkey. Often, however, interface methods are more subtle interaction than explicit *push* or *pull* modalities. For example, a type-ahead feature in a web browser’s URL field might offer ‘wikipedia.org’ after the user types ‘w’, with the recommendation accepted by pressing <enter>. Once the user is familiar with this interaction, they might habitually use ‘w <enter>’ as a shortcut, effectively ‘pulling’ the feature.

The success of the designer’s approach to locus of control is influenced by many factors, including the probability that system feedback accurately reflects the user’s intention, the ease with which causal relationships between action and effect can be learned, their stability and predictability, the temporal connection between action and response, the user’s degree of focus on the work environment, and the potential costs of interruption. Many of these design factors are discussed in the context of autonomous services and intelligent agents under the topic of ‘mixed initiative user interfaces’ [Horvitz 1999].

7.4 User interface customisation and adaptation

User interface customization facilities are an extreme form of user-centered locus of control, where the user changes the interface configuration to better match their needs. Adaptive systems lie at the opposite extreme of the locus of control spectrum, with the interface undergoing *automatic* reconfiguration based on some prediction of improved fit with user needs. Both interface customization and adaptive interfaces are substantial research areas that are beyond the scope of this review. However, some of the primary limitations of customization and adaptation are similar to those discussed above. In particular, Simon's 'satisficing' (Section 2.10) and Carroll's 'paradox of the active user' (Section 1) deter users from customizing their interfaces; and automatic adaptations are likely to cause a temporary performance dip (Section 5) due to the need for the user to identify and comprehend the interface adaptation. Those interested in interface customization and adaptable user interfaces are directed to Bunt et al. [2007] and Gajos et al. [2006].

8. SUMMARY OF RESULTS FROM THE FOUR DOMAINS

The previous sections reviewed four domains of research that explore ways of helping users move from novice to expert performance. The four domains – intramodal improvement, intermodal improvement, vocabulary extension, and task mapping – characterize different opportunities for improving performance, and are derived from a conceptual view of interface interaction as consisting of the user's *performance characteristics* with a set of *alternative methods* for activating a set of *functions* that are required to complete a *task* (see Figure 1).

Intramodal improvement concerns the rapidity and magnitude of performance improvement with one particular interactive method. Key results from this domain include:

- Making an interface method visually salient can improve initial performance. However, long-term performance with advanced methods (e.g., shortcuts) will be impaired if they are removed from the display to raise the prominence of novice methods.
- Memory for interface methods can be improved by inducing greater mental effort about them (prompting deeper encodings, Section 2.4). Doing so may give long-term performance benefits, but at the cost of slower initial performance (also with risks of user frustration). This effect has important implications for the use of *guidance* in assisting novice users complete their tasks, as too much guidance may impair memory development.
- The principle of *rehearsal* states that making physical actions in novice actions as similar as possible to the form of the expert's actions will (i.e., novices rehearse expert behavior) will facilitate skill development.
- Flat command hierarchies with predictable methods for selection (e.g., spatial or syntactic consistency) reduce the number of actions required to select commands, facilitating high performance ceilings.

Intermodal improvement concerns ways to assist users in switching to faster methods for accessing a particular function. Key observations from this domain include:

- Switching to a new interface method is likely to cause a temporary dip in performance. The existence of this dip may deter users from changing methods.
- Interfaces that force users to encounter or use faster methods, or punish them for not using faster methods, can succeed in the lab but may not be acceptable in practice.

- Subtler presentation schemes (such as ‘calm notification’) that show faster methods without demanding attention may be more acceptable to users.
- Encouraging users to reflect about their performance may assist in reducing the tendency to persistently maintain suboptimal methods.

Vocabulary extension considers ways to help users broaden their knowledge and their use of the range of functions available in an interface. Key results from this domain include:

- Most users only know a small subset of the many commands that are available in an interface.
- The conceptual deterrents and barriers to vocabulary extension are similar to those described for intermodal improvement, and many of the same strategies and techniques from that domain can also be used here.
- Two important parts of vocabulary extension are the *generation* of good recommendations, and the *presentation* of those recommendations in the interface.
- Presentation should be dynamically updated to the user’s context, should be continuously available to the user, and should be presented in an ambient manner that allows quick access without interrupting task execution.
- Generation mechanisms should produce recommendations that are both new to the user, and useful given their current context and task.

Task mapping addresses higher-level issues of the strategies that users adopt when seeking to complete their tasks with a user interface. Key findings from this domain include:

- Several novel training approaches have been investigated – for example, gamification can increase engagement and performance; involving the user community can increase the coverage and accuracy of tutorials; and explicitly supporting knowledge transfer can aid users as they switch between systems.
- Knowledge of tasks and tools may not be enough for a user to achieve high performance; supporting users in gaining strategy knowledge is also important.
- The presentation of training information should fit within the user’s current context and should avoid interruptions that break the flow of work.

9. DISCUSSION

This survey focused on interface techniques that are designed to assist users in gaining UI expertise. Underlying human factors that may influence the success of these techniques were briefly reviewed in Section 2. Four domains of interface performance improvement – intramodal, intermodal, vocabulary extension and task mapping – were then introduced in Section 3, and techniques within each of these domains were reviewed through Sections 4-7. Potential avenues for further work were introduced throughout the paper, with particularly promising research opportunities highlighted in the following subsections.

9.1 From lab studies to field studies

Nearly all of the interfaces described in Sections 4-7 were evaluated in controlled laboratory settings, yet several papers observe that lab studies produce more favorable findings than those conducted in the field (e.g., [Duh et al. 2006; Fitchett et al. 2014]). As results from human factors research show that skill acquisition can be influenced by subtle variations, it is highly desirable that the laboratory findings are validated in the field. However, developing systems that can be used in field studies of real work is extremely difficult.

The software that people use for everyday office work is often proprietary, and the lack of access to source code constrains researchers' ability to modify system behavior and monitor its use. Ideally researchers will have a relationship with the software provider, allowing direct access to source code – for example, Autodesk's 'Chronicle' research system has now been released for actual use (Section 7.2).

When source code is unavailable there are often application programming interfaces (APIs) that allow limited control of software systems running on a particular platform. Typically, these APIs form part of the accessibility framework that assists development of software for users with special needs. Several researchers describe tools that assist with monitoring and logging interaction with unaltered software systems (e.g., [Kukreja et al. 2007; Alexander et al. 2008; Morgan et al. 2013]), and these methods can also be extended to enable some control over proprietary systems, as demonstrated by the Blur and Skillometer systems (Section 5.1), which ran on Microsoft Windows and Apple Mac OSX. However, these techniques have several limitations, including access to only a subset of controls, reduced system responsiveness, and increased frequency of software crashes, which make them impractical for studies of real use.

Open source systems permit modifications enabling longitudinal analysis of expert use (e.g., Murphy et al.'s [2006] analysis of Java developers using the Eclipse IDE), but for productivity applications, the user base is often smaller than major commercial systems.

9.2 Understanding the costs and benefits of intermodal methods

Methods for intermodal improvement, such as keyboard accelerators, are already commonly used in state-of-the-art interfaces. While researchers have examined methods to minimize the intermodal performance dip (Figure 7), there are opportunities for more broadly examining the costs associated with providing more than one method for activating a function. For example, Olson and Nilsen [1988] showed that users who knew two methods for entering formulae into a spreadsheet were significantly slower than users who knew only one, and they attributed the time cost to the additional cognition required to plan and decide between methods. Similarly, Quinn et al. [2013] recently demonstrated that the decision costs associated with choosing between alternative cursor pointing paths negated the benefits of a shortcut pointing method that allowed users to wrap the cursor around display edges.

9.3 Rehearsal and guidance interfaces

As described in Section 4.2, Kurtenbach's [1993] marking menus were designed so that the novice's motor actions were a physical rehearsal of those used when expert. Recently, several researchers have generalized the principle of rehearsal in other forms of interaction, including dragged finger gestures [Bau and Mackay 2008] and hotkeys [Malacria et al. 2013]. It seems likely that the principles of rehearsal can be deployed in other forms of interaction to facilitate expertise development.

For example, FastTap [Gutwin et al. 2014] is a recently proposed interactive technique for improving command selections on tablets – when novice, the user's selections are visually guided by the display of a spatially stable grid of menu items, but once expert multiple selections can be made by chorded taps on the known locations without the need to wait for visual guidance.

As well as devising new interface techniques based on the principle of rehearsal, there are extensive research opportunities in better understanding how interfaces can best support users through different forms of guidance. Anderson and Bischof's [2013] study (Section 4.2) provides a recent example, demonstrating that progressive removal of guidance has learning advantages over its continual availability.

9.4 Deliberate practice, its timing, content and acceptability

Section 2.3 reviewed Ericsson's research, emphasizing the role of deliberate practice in the development of elite levels of skill. Section 2.4 then reviewed the connection between the depth of mental engagement in activities and their resultant memorization. Section 2.6 also reviewed how the distribution of practice sessions influences learning.

There are relatively few examples of user interfaces that explicitly draw on these factors to promote skill development with the interface; consequently, there are many opportunities for further work. These include the following: work on algorithms for identifying potentially beneficial methods, functions, or strategies that the user appears to be unaware of, as well as determining appropriate times and methods for their presentation – for example, Li et al.'s [2011] collaborative filtering algorithms, Section 6); user interface techniques for presenting engaging practice sessions to users – for example, gamification (Section 7.1); and research on how to optimize the level of effort required across various tasks and skill levels (e.g., an adaptive form of the frost brushing interface described in Section 4.2, Figure 4).

9.5 Speed/accuracy/quality tradeoffs across expertise

Most of this review has focused on task performance time, but accuracy and the cost of errors are often critical to successful interaction, and they can vary substantially between interaction techniques. For example, a missed pointer-based click on Microsoft Word's 'bold' icon is likely to be quickly identified by the absence of icon highlighting, and correcting the error will be fast because the cursor is already near the target. In contrast, a user who mistakenly believes the hotkey for 'bold' is <Ctrl-L> will need to infer their error from the change of display state, and then undo the error, home their hands to the mouse, point to the 'bold' icon, dwell to view the tooltip, memorize the hotkey, home their hands back to the keyboard, and activate the hotkey. Research is needed to better understand the role of accuracy and error correction in expertise development. Related research questions arise when considering the quality of the work product rather than outright task time (e.g., the quality of a design for a designer working with a CAD package).

9.6 Interface methods for notification, and their underlying algorithms

Commercial and research user interfaces have made substantial improvements to interaction through increased use of 'mixed initiative' interactions [Bunt et al. 2007], in which the system offers non-intrusive notifications that aim to assist the task during execution (such as URL type-ahead, Section 4.2). Four directions for subsequent research on mixed initiative interactions are: 1. examining how to use new sources of information for generating recommendations, such as crowd-sourcing help recommendations [Chilana et al. 2013]; 2. improved algorithms for generating recommendations, such as AccessRank [Fitchett and Cockburn 2012]; 3. clarifying the relationship between post-action feedback and during-action feedforward (e.g., Vermeulen, Luyten, Hoven and Coninx [2013]); and 4. new mixed initiative interactions (e.g., Octopocus, Section 4.2).

9.7 Generalisability of previous results

Human factors literature, summarized in Section 2, shows that many factors influence skill acquisition. Interaction with computing systems introduces new and important factors that have not been extensively studied. It is therefore important that research findings are validated in varied settings, with diverse user groups and over prolonged periods. Factors that might influence results include different input and output configurations, and how these configurations interact with task factors. For example, it is common for users in high workload situations (such as air traffic

control, stock trading, or emergency dispatch) to use multiple displays to monitor concurrent activities. While there has been research on how highly trained experts coordinate their activities in such settings [Hornof et al. 2010] there is scope for research into skill acquisition in such settings. Other factors that might be examined include *user factors* (culture, stress, age, etc.), *task factors* (e.g., high cognitive demand versus low, workload, task pace), *input device* (e.g., touchscreen, keyboard, mouse, trackpad), and *output configuration* (e.g., the size and location of data on single or multiple displays, and the effect of output modalities such as sound and haptic feedback).

10. CONCLUSIONS

Graphical user interfaces have become the primary medium through which office work is conducted. A primary factor contributing to their success is the ease with which they can be learned – newcomers can quickly attain sufficient competence to adequately complete their work. While ease of learning is highly desirable, it is common for office workers to continue using substantially similar interfaces for months, years, and decades, so optimizing designs for initial learnability rather than long-term efficiency can be counterproductive. Current interface guidelines attempt to account for novice learning *and* expertise by advocating for shortcut facilities, but this strategy is known to fail as few users make the transition to expert methods – instead, users maintain the adequate but inefficient strategies they learned first.

There has been a recent surge of research interest in resolving these problems of graphical user interfaces tending to trap users in a beginner mode of operation. While some of these research projects are founded on established findings from the human factors literature, others are not. This paper provided a review of key findings from the human factors literature that are relevant to supporting transitions from novice to expert performance with user interfaces, as well as summarizing the state of the art in interface research supporting the transition. We provided directions for further work and look forward to a new generation of user interfaces that facilitate expertise as well as ease of use.

ACKNOWLEDGEMENTS

This work was partially funded by Royal Society of New Zealand Marsden Grant 10-UOC-020. Thanks to the anonymous reviewers and Associate Editor Jonathan Grudin for their comments and recommendations on earlier drafts of this paper.

REFERENCES

- ACKERMAN, P. L. 2007. New Developments in Understanding Skilled Performance. *Current Directions in Psychological Science* **16**(5): 235-239.
- ADAR, E., TAN, D. S. AND TEEVAN, J. 2013. Benevolent deception in human computer interaction. *Proceedings of the ACM Conference on Human Factors in Computing Systems, Paris, France, 1863-1872*. ACM.
- ALEXANDER, J. 2009. *Understanding and Improving Navigation within Electronic Documents*. Computer Science and Software Engineering. Christchurch, University of Canterbury. **PhD**.
- ALEXANDER, J., COCKBURN, A. AND LOBB, R. 2008. AppMonitor: A Tool for Recording User Actions in Unmodified Windows Applications. *Behavior Research Methods* **40**(2): 413-421.
- ANDERSON, F. AND BISCHOF, W. F. 2013. Learning and performance with gesture guides. *Proceedings of the ACM Conference on Human Factors in Computing Systems, Paris, France, 1109-1118*. ACM.
- ANDERSON, J. 1995. *Learning and Memory*. New York, John Wiley.
- ANDERSON, J. 2005. *Cognitive Psychology and its Implications*. New York, Worth Publishers.
- APPERT, C. AND BAU, O. 2010. Scale detection for a priori gesture recognition. *Proceedings of the ACM Conference on Human Factors in Computing Systems, Atlanta, Georgia, USA, 879-882*. ACM.

- ARMSTRONG, T. 1970. Training for the production of memorized movement patterns. Ann Arbor, MI, University of Michigan, Department of Psychology.
- BADDELEY, A. AND LONGMAN, D. 1978. The influence of length and frequency of training session on the rate of learning to type. *Ergonomics* **21**: 627-635.
- BADDELEY, A. D. 1999. *Essentials of Human Memory*. Hove, England, Psychology Press.
- BANOVIC, N., CHEVALIER, F., GROSSMAN, T. AND FITZMAURICE, G. 2012. Triggering triggers and burying barriers to customizing software. Proceedings of the ACM Conference on Human Factors in Computing Systems, Austin, Texas, USA, 2717-2726. ACM.
- BARRETT, R., KANDOGAN, E., MAGLIO, P., HABER, E., TAKAYAMA, L. AND PRABAKER, M. 2004. Field studies of computer system administrators: analysis of system management tools and practices. Proceedings of the ACM Conference on Computer Supported Cooperative Work, Chicago, Illinois, USA, 388-395. ACM Press.
- BATEMAN, S., TEEVAN, J. AND WHITE, R. W. 2012. The search dashboard: how reflection and comparison impact search behavior. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Austin, Texas, USA, 1785-1794. ACM.
- BAU, O. AND MACKAY, W. E. 2008. OctoPocus: a dynamic guide for learning gesture-based command sets. Proceedings of the ACM Symposium on User Interface Software and Technology, Monterey, CA, USA, 37-46. ACM.
- BENNETT, M., MCCARTHY, K., O'MODHRAN, S. AND SMYTH, B. 2011. Simpleflow: enhancing gestural interaction with gesture prediction, abbreviation and autocompletion. Proceedings of the 13th IFIP TC 13 International Conference on Human-Computer Interaction - Volume Part I, Lisbon, Portugal, 591-608. Springer-Verlag.
- BHAVNANI, S. AND JOHN, B. 2000. The Strategic Use of Complex Computer Systems. *Human Computer Interaction* **15**: 107-137.
- BHAVNANI, S. K. AND JOHN, B. E. 1996. Exploring the unrealized potential of computer-aided drafting. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Vancouver, British Columbia, Canada, ACM.
- BODKER, S. 1995. Applying activity theory to video analysis: how to make sense of video data in human-computer interaction. Context and consciousness, Massachusetts Institute of Technology: 147-174.
- BOURNE, L. AND ARCHER, E. 1956. Time continuously on target as a function of distribution of practice. *Journal of Experimental Psychology* **51**: 25-33.
- BULLOCK, W. A. AND GILLILAND, K. 1993. Eysenck's arousal theory of introversion/extraversion: A converging measures investigation. *Journal of Personality and Social Psychology* **64**(1): 113-123.
- BUNT, A., CONATI, C. AND MCGRENERE, J. 2007. Supporting Interface Customization using a Mixed-Initiative Approach. Proceedings of IUI '07: Conference on Intelligent User Interfaces, Honolulu, Hawaii, 92-101. ACM.
- CARD, S. K., MORAN, T. P. AND NEWELL, A. 1983. *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates.
- CARROLL, J. 1987. The adventure of getting to know a computer. *Readings in Human-Computer Interaction: A Multidisciplinary Approach*. BAECKER, R. AND BUXTON, W., Morgan Kaufmann Publishers Inc. : 639-648.
- CARROLL, J. AND CARRITHERS, C. 1984. Training Wheels in a User Interface. *Communications of the ACM* **27**(8): 800-806.
- CARROLL, J. AND ROSSEN, M. 1987. Paradox of the active user. *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*. CARROLL, J. Cambridge, MA, MIT Press: 80-111.
- CEPEDA, N., PASHLER, H., VUL, E., WIXTED, J. AND ROHRER, D. 2006. Distributed Practice in Verbal Recall Tasks: A Review and Quantitative Synthesis. *Psychological Bulletin* **132**(3): 354-380.
- CHILANA, P. K., KO, A. J., WOBROCK, J. O. AND GROSSMAN, T. 2013. A multi-site field study of crowdsourced contextual help: usage and perspectives of end users and software teams. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Paris, France, 217-226. ACM.
- COCKBURN, A. AND GUTWIN, C. 2009. A Predictive Model of Human Performance with Scrolling and Hierarchical Lists. *Human-Computer Interaction* **24**(3): 273-314.

COCKBURN, A., GUTWIN, C. AND GREENBERG, S. 2007. A Predictive Model of Menu Performance. Proceedings of CHI'07: ACM Conference on Human Factors in Computing Systems, San Jose, CA, 627-636. ACM Press.

COCKBURN, A., KRISTENSSON, P., ALEXANDER, J. AND ZHAI, S. 2007. Hard Lessons: Effort-Inducing Interfaces Benefit Spatial Learning. Proceedings of CHI'07: ACM Conference on Human Factors in Computing Systems, San Jose, CA, 1571-1580. ACM Press.

COCKBURN, A. AND MCKENZIE, B. 2002. Evaluating the Effectiveness of Spatial Memory in 2D and 3D Physical and Virtual Environments. Proceedings of CHI'2002 Conference on Human Factors in Computing Systems, Minneapolis, Minnesota, 203-210. ACM Press.

CRAIK, F. AND LOCKHART, R. 1972. Levels of processing: A framework for memory research. *Journal of Verbal Learning and Verbal Behavior* **11**: 671-684.

CRAIK, F. AND TULVING, E. 1975. Depth of processing and the retention of words in episodic memory. *Journal of Experimental Psychology: General* **103**(3): 268-294.

CROSSMAN, E. 1959. A theory of acquisition of speed-skill. *Ergonomics* **2**(2): 153-166.

CZERWINSKI, M., HORVITZ, E. AND CUTTRELL, E. 2001. Subjective Duration Assessment: An Implicit Probe for Software Usability. Proceedings of IHM-HCI, Lille, France, 167-170.

DELANEY, P. F., REDER, L. M., STASZEWSKI, J. J. AND RITTER, F. E. 1998. The Strategy-Specific Nature of Improvement: The Power Law Applies by Strategy Within Task. *Psychological Science* **9**(1): 1-7.

DETERDING, S., DIXON, D., KHALED, R. AND NACKE, L. 2011. From game design elements to gamefulness: defining "gamification". Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, Tampere, Finland, 9-15. ACM.

DIEWALD, S., MOLLER, A., ROALTER, L., STOCKINGER, T. AND KRANZ, M. 2013. Gameful design in the automotive domain: review, outlook and challenges. Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications, Eindhoven, Netherlands, 262-265. ACM.

DIX, A., FINLAY, J., ABOWD, G. AND BEALE, R. 1993. *Human-Computer Interaction*. Harlow, England, Prentice Hall.

DOANE, S., PELLEGRINO, J. AND KLATZKY, R. 1990. Expertise in a Computer Operating System: Conceptualization and Performance. *Human Computer Interaction* **5**: 267-304.

DONG, T., DONTCHEVA, M., JOSEPH, D., KARAHALIOS, K., NEWMAN, M. AND ACKERMAN, M. 2012. Discovery-based games for learning software. Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems, Austin, Texas, USA, 2083-2086. ACM.

DRAPER, S. 1984. The Nature of Expertise in Unix. INTERACT 84: 1st IFIP International Conference on Human-Computer Interaction, London, UK, 465-471. North-Holland.

DUH, H. B.-L., TAN, G. C. B. AND CHEN, V. H.-H. 2006. Usability evaluation for mobile device: a comparison of laboratory and field tests. Proceedings of the 8th conference on Human-computer interaction with mobile devices and services, Helsinki, Finland, 181-186. ACM.

DYCK, J., PIENELLE, D., BROWN, B. AND GUTWIN, C. 2003. Learning from Games: HCI Innovations in Entertainment Software. Proceedings of Graphics Interface, Halifax, 237-246. AK Peters.

EKSTRAND, M., LI, W., GROSSMAN, T., MATEJKA, J. AND FITZMAURICE, G. 2011. Searching for software learning resources using application context. Proceedings of the 24th ACM Symposium on User Interface Software and Technology, Santa Barbara, California, USA, 195-204. ACM.

ENGSTRÖM, J., JOHANSSON, E. AND ÖSTLUND, J. 2005. Effects of visual and cognitive load in real and simulated motorway driving. *Transportation Research Part F: Traffic Psychology and Behaviour* **8**(2): 97-120.

ERICSSON, K. 2004. Deliberate Practice and the Acquisition and Maintenance of Expert Performance in Medicine and Related Domains. *Academic Medicine* **79**(10): S1-S12.

ERICSSON, K., PRIETULA, M. AND COKELY, E. 2007. The Making of an Expert. *Harvard Business Review* **July-August**: 1-7.

ERICSSON, K. A. 1998. The Scientific Study of Expert Levels of Performance: general implications for optimal learning and creativity 1. *High Ability Studies* **9**(1): 75--100.

FINDLATER, L., MOFFATT, K., MCGRENERE, J. AND DAWSON, J. 2009. Ephemeral adaptation: the use of gradual onset to improve menu selection performance. Proceedings CHI'09 Conference on Human Factors in Computing Systems, Boston, MA, 1655-1664. ACM Press.

FISCHER, G. 2001. User Modeling in Human-Computer Interaction. *User Modeling and User-Adapted Interaction* **11**(1/2): 65--86.

FITCHETT, S. AND COCKBURN, A. 2012. AccessRank: predicting what users will do next. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Austin, Texas, USA, 2239-2242. ACM.

FITCHETT, S., COCKBURN, A. AND GUTWIN, C. 2014. Finder Highlights: Field Evaluation and Design of an Augmented File Browser. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Paris, France, 3685-3694. ACM.

FITTS, P. 1964. Perceptual-motor skills learning. *Categories of human learning*. MELTON, A. New York, Academic Press: 243-285.

FITTS, P. M. AND POSNER, M. I. 1967. *Human Performance*. Belmont, CA, Brookes/Cole.

FLATLA, D. R., GUTWIN, C., NACKE, L. E., BATEMAN, S. AND MANDRYK, R. L. 2011. Calibration games: making calibration tasks enjoyable by adding motivating game elements. *Proceedings of the 24th ACM Symposium on User Interface Software and Technology*, Santa Barbara, California, USA, 403-412. ACM.

FREEMAN, D., BENKO, H., MORRIS, M. R. AND WIGDOR, D. 2009. ShadowGuides: visualizations for in-situ learning of multi-touch and whole-hand gestures. *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, Banff, Alberta, Canada, 165-172. ACM.

FU, W.-T. AND GRAY, W. D. 2004. Resolving the paradox of the active user: stable suboptimal performance in interactive tasks. *Cognitive Science* **28**(6): 901-935.

GAJOS, K., CZERWINSKI, M., TAN, D. AND WELD, D. 2006. Exploring the Design Space for Adaptive Graphical User Interfaces. *Proceedings of AVI'06: ACM Conference on Advanced Visual Interfaces*, Venice, Italy, 201-208. ACM Press.

GENTILE, A. 1972. A Working Model of Skill Acquisition. *Quest* **17**: 3-23.

GRAY, W. D., SIMS, C., FU, W.-T. AND SCHOELLES, M. 2006. The soft constraints hypothesis: A rational analysis approach to resource allocation for interactive behavior. *Psychological Review* **113**: 461-482.

GREENBERG, S. AND WITTEN, I. 1993. Supporting Command Reuse: Empirical Foundations and Principles. *International Journal of Man-Machine Studies* **39**: 353-390.

GROSSMAN, T., DRAGICEVIC, P. AND BALAKRISHNAN, R. 2007. Strategies for Accelerating On-line Learning of Hotkeys. *Proceedings of CHI'07: ACM Conference on Human Factors in Computing Systems*, San Jose, California, 1591-1600. ACM Press.

GROSSMAN, T., MATEJKA, J. AND FITZMAURICE, G. 2010. Chronicle: capture, exploration, and playback of document workflow histories. *Proceedings of the 23rd ACM Symposium on User Interface Software and Technology*, New York, New York, USA, 143-152. ACM.

GUTWIN, C., COCKBURN, A., SCARR, J. AND MALACRIA, S. 2014. Faster Command Selection on Tablets with FastTap. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Paris, France, 2617-2626. ACM.

HASEGAWA, M., CARPENTER, P. A. AND JUST, M. A. 2002. An fMRI Study of Bilingual Sentence Comprehension and Workload. *NeuroImage* **15**(3): 647-660.

HEATHCOTE, A., BROWN, S. AND MEWHORT, D. 2000. The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin and Review* **7**(2): 185-207.

HENDY, J., BOOTH, K. AND MCGRENERE, J. 2010. Graphically Enhanced Keyboard Accelerators for GUIs. *Proceedings of Graphics Interface*, Ottawa, Ontario, 3-10

HEWSON, M. AND LITTLE, M. 2001. Giving Feedback in Medical Education. *Journal of General Internal Medicine* **13**(2): 111-116.

HORNOF, A. J., ZHANG, Y. AND HALVERSON, T. 2010. Knowing where and when to look in a time-critical multimodal dual task. *Proceedings of the 28th SIGCHI Conference on Human Factors in Computing Systems*, Atlanta, Georgia, USA, 2103-2112. ACM.

HORNOF, A. J. AND KIERAS, D. E. 1997. Cognitive Modeling Reveals Menu Search is Both Random and Systematic. *Proceedings of CHI'97: ACM Conference on Human Factors in Computing Systems*, Atlanta, Georgia, 107-114.

HORVITZ, E. 1999. Principles of mixed-initiative user interfaces. *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, Pittsburgh, Pennsylvania, USA, 159-166. ACM.

HSI, I. AND POTTS, C. 2000. Studying the Evolution and Enhancement of Software Features. Proceedings of the International Conference on Software Maintenance (ICSM'00), 143. IEEE Computer Society.

HYÖNÄ, J., TOMMOLA, J. AND ALAJA, A.-M. 1995. Pupil Dilation as a Measure of Processing Load in Simultaneous Interpretation and Other Language Tasks. The Quarterly Journal of Experimental Psychology Section A **48**(3): 598-612.

KAHNEMAN, D. 1973. Attention and Effort. New-Jersey, Prentice-Hall Inc.

KARAT, J., KARAT, C. AND UKELSON, J. 2000. Affordances, motivation and the design of user interfaces. Communications of the ACM **43**(8): 49-51.

KELLEHER, C. AND PAUSCH, R. 2005. Stencils-based tutorials: design and evaluation. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Portland, Oregon, USA, 541-550. ACM.

KIM, J. AND RITTER, F. 2013. Learning, Forgetting, and Relearning for Keystroke- and Mouse-Driven Tasks: Relearning is Important. Human-Computer Interaction: dx.doi.org/10.1080/07370024.2013.828564.

KONG, N., GROSSMAN, T., HARTMANN, B., AGRAWALA, M. AND FITZMAURICE, G. 2012. Delta: a tool for representing and comparing workflows. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Austin, Texas, USA, 1027-1036. ACM.

KRISLER, B. AND ALTERMAN, R. 2008. Training towards mastery: overcoming the active user paradox. Proceedings of the 5th Nordic Conference on Human-Computer Interaction, Lund, Sweden, 239-248. ACM.

KRISTENSSON, P. O. AND DENBY, L. C. 2011. Continuous recognition and visualization of pen strokes and touch-screen gestures. Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling, Vancouver, British Columbia, Canada, 95-102. ACM.

KUKREJA, U., STEVENSON, W. AND RITTER, F. 2007. RUI: Recording User Input from Interfaces Under Windows and Mac OS X. Behavior Research Methods, Instruments, and Computers **38**(4): 656-659.

KURTENBACH, G. 1993. The Design and Evaluation of Marking Menus. Department of Computer Science. Toronto, University of Toronto. **Ph.D.**

KURTENBACH, G. AND BUXTON, B. 1993. The Limits of Expert Performance Using Hierarchic Marking Menus. Proceedings of InterCHI'93, 482-487.

KURTENBACH, G. AND BUXTON, W. 1994. User Learning and Performance with Marking Menus. Proceedings of CHI'94 Conference on Human Factors in Computing Systems, Boston, 258-264. ACM.

KURTENBACH, G., MORAN, T. AND BUXTON, W. 1994. Contextual Animation of Gestural Commands. Computer Graphics Forum **13**(5): 305-314.

KURTENBACH, G., SELLEN, A. AND BUXTON, W. 1993. An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus. Human-Computer Interaction **8**(1): 1-23.

KYLLO, L. B. AND LANDERS, D. M. 1995. Goal setting in sport and exercise: A research synthesis to resolve the controversy. Journal of Sport & Exercise Psychology **17**(2): 117-137.

LAFRENIERE, B., GROSSMAN, T. AND FITZMAURICE, G. 2013. Community Enhanced Tutorials: Improving Tutorials with Multiple Demonstrations. Proceedings of CHI'13: ACM Conference on Human Factors in Computing Systems, Paris, France, 1779-1788. ACM.

LANE, D. M., NAPIER, H. A., PERES, S. C. AND SANDOR, A. 2005. Hidden costs of graphical user interfaces: Failure to make the transition from menus and icon toolbars to keyboard shortcuts. International Journal of Human Computer Interaction **18**(2): 133-144.

LI, W., GROSSMAN, T. AND FITZMAURICE, G. 2012. GamiCAD: a gamified tutorial system for first time autocad users. Proceedings of the 25th annual ACM symposium on User interface software and technology, Cambridge, Massachusetts, USA, 103-112. ACM.

LI, W., MATEJKA, J., GROSSMAN, T., KONSTAN, J. A. AND FITZMAURICE, G. 2011. Design and evaluation of a command recommendation system for software applications. ACM Trans. Comput.-Hum. Interact. **18**(2): 1-35.

LINEHAN, C., KIRMAN, B., LAWSON, S. AND CHAN, G. 2011. Practical, appropriate, empirically-validated guidelines for designing educational games. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Vancouver, BC, Canada, 1979-1988. ACM.

LOCKE, E. AND LATHAM, G. 2006. New Directions in Goal-Setting Theory. *Current Directions in Psychological Science* **15**: 265-268.

MALACRIA, S., BAILLY, G., HARRISON, J., COCKBURN, A. AND GUTWIN, C. 2013. Promoting Hotkey Use through Rehearsal with ExposeHK. *Proceedings of CHI'13: ACM Conference on Human Factors in Computing Systems, Paris, France*, 573-582. ACM.

MALACRIA, S., SCARR, J., COCKBURN, A., GUTWIN, C. AND GROSSMAN, T. 2013. Skillometers: Reflective Widgets that Motivate and Help Users to Improve Performance. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '13), St. Andrews, Scotland*, 321-330. ACM.

MALONE, T. 1980. What makes things fun to learn? A study of intrinsically motivating computer games. Department of Psychology. Palo Alto, CA, Stanford University.

MATEJKA, J., GROSSMAN, T. AND FITZMAURICE, G. 2011. Ambient Help. *Proceedings of CHI'11: ACM Conference on Human Factors in Computing Systems, Vancouver, Canada*, 2751-2760. ACM Press.

MATEJKA, J., GROSSMAN, T. AND FITZMAURICE, G. 2013. Patina: Dynamic Heatmaps for Visualizing Application Usage. *Proceedings of CHI'13: ACM Conference on Human Factors in Computing Systems, Paris, France*, 3227-3236. ACM.

MATEJKA, J., LI, W., GROSSMAN, T. AND FITZMAURICE, G. 2009. CommunityCommands: Command recommendations for software applications. *ACM Symposium on User Interface Software and Technology*, 193-202. ACM.

MCGONIGAL, J. 2011. *Reality is Broken: Why Games Make Use Better and How They Can Change the World*, Penguin Group.

MCGRENERE, J., BAECKER, R. M. AND BOOTH, K. S. 2007. A field evaluation of an adaptable two-interface design for feature-rich software. *ACM Trans. Comput.-Hum. Interact.* **14**(1): 3.

MORGAN, J. H., CHENG, C.-Y., PIKE, C. AND RITTER, F. E. 2013. A Design, Tests and Considerations for Improving Keystroke and Mouse Loggers. *Interacting with Computers* **25**(3): 242-258.

MURPHY, G. C., KERSTEN, M. AND FINDLATER, L. 2006. How Are Java Software Developers Using the Eclipse IDE? *IEEE Softw.* **23**(4): 76-83.

NAVEH-BENJAMIN, M. 1987. Recognition memory of spatial location information: Another failure to support automaticity. *Memory and Cognition* **16**: 437-445.

NEWELL, A. AND ROSENBLOOM, P. S. 1981. Mechanisms of Skill Acquisition and the Law of Practice. *Cognitive Skills and their Acquisition*. ANDERSON, J. Hillsdale, NJ, Erlbaum: 1-55.

NIELSEN, J. 1993. *Usability Engineering*. San Francisco, Morgan Kaufmann.

NILSEN, E., JONG, H., OLSON, J. S., BIOLSI, K., RUETER, H. AND MUTTER, S. 1993. The growth of software skill: a longitudinal look at learning and performance. *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems, Amsterdam, The Netherlands*, 149-156. ACM.

NORMAN, D. 1983. Design principles for Human-Computer Interaction. *Proceedings of CHI'83 SIGCHI Conference on Human Factors in Computing Systems, Boston, MA*, 1-20. ACM Press.

NORMAN, D. 1983. Design principles for Human-Computer Interfaces. *Proceedings of CHI 83*: 1--10.

NOVICK, D. G. AND WARD, K. 2006. Why don't people read the manual? *Proceedings of the 24th Annual ACM International Conference on Design of Communication, Myrtle Beach, SC, USA*, 11-18. ACM.

ODELL, D. L. , DAVIS, R., C., SMITH, A. AND WRIGHT, P., K. 2004. Toolglasses, marking menus, and hotkeys: a comparison of one and two-handed command selection techniques. *Proceedings of Graphics Interface 2004, London, Ontario, Canada*, 17-24. Canadian Human-Computer Communications Society.

OLSON, J. R. AND NILSEN, E. L. 1988. Analysis of Cognition Involved in Spreadsheet Software Interaction. *Human Computer Interaction* **3**(4): 309-350.

PERES, S., TAMBORELLO, F., FLEETWOOD, M., CHUNG, P. AND PAIGE-SMITH, D. 2004. Keyboard shortcut usage: The roles of social factors and computer experience. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 803-807.

PERKINS-CECCATO, N., PASSMORE, S. AND LEE, T. 2003. Effects of focus of attention depend on golfers' skill. *Journal of Sports Sciences* **21**: 593-600.

PLIMMER, B., CROSSAN, A., BREWSTER, S. A. AND BLAGOJEVIC, R. 2008. Multimodal collaborative handwriting training for visually-impaired people. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Florence, Italy, 393-402. ACM.

QUINN, P., COCKBURN, A. AND DELAMARCHE, J. 2013. Examining the costs of multiple trajectory pointing techniques. *International Journal of Human-Computer Studies* **71**(4): 492-509.

RAMESH, V., HSU, C., AGRAWALA, M. AND HARTMANN, B. 2011. ShowMeHow: translating user interface instructions between applications. *Proceedings of the 24th ACM Symposium on User Interface Software and Technology*, Santa Barbara, California, USA, 127-134. ACM.

RANDALL, N. AND PEDERSEN, I. 1998. Who exactly is trying to help us? The ethos of help systems in popular computer applications. *Proceedings of the 16th Annual International Conference on Computer Documentation*, Quebec, Quebec, Canada, 63-69. ACM.

REEVES, B. AND READ, J. 2009. *Total Engagement: Using Games and Virtual Worlds to Change the Way People Work and Businesses Compete*, Harvard Business Press.

REVELLE, W., HUMPHREYS, M., SIMON, L. AND GILLILAND, K. 1980. The Interactive Effect of Personality, Time of Day, and Caffeine: A Test of the Arousal Model. *Journal of Experimental Psychology* **109**(1): 1-31.

RHIENMORA, P., GAJANANAN, K., HADDAWY, P., SUEBNUKARN, S., DAILEY, M. N., SUPATARATARN, E. AND SHRESTHA, P. 2010. Haptic augmented reality dental trainer with automatic performance assessment. *Proceedings of the 15th International Conference on Intelligent User Interfaces*, Hong Kong, China, 425-426. ACM.

RIEMAN, J. 1996. A field study of exploratory learning strategies. *ACM Trans. Comput.-Hum. Interact.* **3**(3): 189-218.

ROSSON, M. B. 1983. Patterns of experience in text editing. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Boston, Massachusetts, USA, 171-175. ACM.

ROTHSTEIN, A. AND ARNOLD, R. 1976. Bridging the gap: Application of research on videotape feedback and bowling. *Motor Skills: Theory Into Practice* **1**: 35-62.

SALTHOUSE, T. 1985. Anticipatory processes in transcription typing. *Journal of Applied Psychology* **70**: 264-271.

SCARR, J., COCKBURN, A., GUTWIN, C. AND BUNT, A. 2012. Improving command selection with CommandMaps. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Austin, Texas, USA, 257-266. ACM.

SCARR, J., COCKBURN, A., GUTWIN, C. AND MALACRIA, S. 2013. Testing the robustness and performance of spatially consistent interfaces. *Proceedings of the 2013 SIGCHI Conference on Human Factors in Computing Systems*, Paris, France, 3139-3148. ACM.

SCARR, J., COCKBURN, A., GUTWIN, C. AND QUINN, P. 2011. Dips and Ceilings: Understanding and Supporting Transitions to Expertise in User Interfaces. *Proceedings of CHI'11: ACM Conference on Human Factors in Computing Systems*, Vancouver, Canada, 2741-2750. ACM Press.

SCHMIDT, R. 1991. Frequent augmented feedback can degrade learning: Evidence and interpretations. *Tutorials in motor neuroscience*. REQUIN, J. AND STELMACH, G.: 59-75.

SCHMIDT, R. AND BJORK, R. 1992. The Conceptualizations of Practice: Common Principles in Three Paradigms Suggest New Concepts for Training. *Psychological Science* **3**(4): 207-217.

SCHMIDT, R. AND LEE, T. 2011. *Motor Control and Learning: A Behavioral Emphasis*. New York, Human Kinetics.

SCHNEIDER, W. AND SHIFFRIN, R. 1977. Controlled and automatic human information processing: I. Detection, search, and attention. *Psychological Review* **84**(1): 1-66.

SHNEIDERMAN, B. 1987. *Direct Manipulation: A Step Beyond Programming Languages* (excerpt). *Readings in Human-Computer Interaction: A Multidisciplinary Approach*. BAECKER, R. AND BUXTON, W., Morgan-Kauffman: 461-467.

SHNEIDERMAN, B. 1992. *Designing the User Interface*, Addison-Wesley.

SHNEIDERMAN, B. 2003. Promoting universal usability with multi-layer interface design. *Proceedings of the conference on Universal Usability*, Vancouver, Canada, 1-8. ACM Press.

SHROYER, R. 2000. Actual readers versus implied readers: role conflicts in Office 97. *Journal of the Society for Technical Communication* **47**(2): 238-40.

- SIMON, H. 1959. Theories of Decision-Making in Economics and Behavioral Science. *The American Economic Review* **49**(3): 252-283.
- SNODDY, G. 1926. Learning and stability: A psychophysical analysis of a case of motor learning with clinical applications. *Journal of Applied Psychology* **10**: 1-36.
- TAK, S., WESTENDORP, P. AND VAN ROOIJ, I. 2013. Satisficing and the use of keyboard shortcuts: Being good enough is enough? *Interacting with Computers* **16**(1): 16.
- TAN, D. S., STEFANUCCI, J. K., PROFFITT, D. R. AND PAUSCH, R. 2001. The InfoCockpit: providing location and place to aid human memory. *Proceedings of the 2001 Workshop on Perceptive User Interfaces*, Orlando, Florida, 1-4. ACM.
- TAUSCHER, L. AND GREENBERG, S. 1997. How People Revisit Web Pages: Empirical Findings and Implications for the Design of History Systems. *International Journal of Human Computer Studies*, Special Issue on World Wide Web Usability **47**(1): 97-138.
- THOMAS, G. 2013. *Education: A Very Short Introduction*, Oxford University Press.
- TUBBS, M. 1986. Goal setting: A meta-analytic examination of empirical evidence. *Journal of Applied Psychology* **71**(3): 474-483.
- VAN ASSELEN, M., FRITSCHY, E. AND POSTMA, A. 2005. The influence of intentional and incidental learning on acquiring spatial knowledge during navigation. *Psychological Research* **70**(2): 151-156.
- VERMEULEN, J., LUYTEN, K., HOVEN, E. V. D. AND CONINX, K. 2013. Crossing the bridge over norman's gulf of execution: revealing feedforward's true identity. *Proceedings of the 2013 SIGCHI Conference on Human Factors in Computing Systems*, Paris, France, 1931-1940. ACM.
- WALLACE, S. A. AND HAGLER, R. W. 1979. Knowledge of Performance and the Learning of a Closed Motor Skill. *Research Quarterly. American Alliance for Health, Physical Education, Recreation and Dance* **50**(2): 265-271.
- WHITESIDE, J., JONES, S., LEVY, P. AND WIXON, D. 1985. User Performance with Command, Menu, and Iconic Interfaces. *Proceedings of CHI '85: ACM Conference on Human Factors in Computing Systems*, 185-191. ACM Press.
- WOLFE, J. 1998. What Can 1 Million Trials Tell Us about Visual Search. *Psychological Science* **9**(1): 33-97.
- WULF, G. AND SHEA, C. 2002. Principles derived from the study of simple skills do not generalize to complex skill learning. *Psychonomic Bulletin and Review* **9**(2): 185-211.
- YECHIAM, E., EREV, I., YEHENE, V. AND GOPHER, D. 2003. Melioration and the transition from touch-typing training to everyday use. *Human Factors* **45**(4): 671-684.
- ZHAI, S. AND KRISTENSSON, P. 2003. Shorthand Writing on Stylus Keyboard. *Proceedings of CHI'2003 Conference on Human Factors in Computing Systems*, Fort Lauderdale, Florida, 97-104.
- ZHAI, S., SUE, A. AND ACCOT, J. 2002. Movement model, hits distribution and learning in virtual keyboarding. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Minneapolis, Minnesota, USA, 17-24. ACM.