Learning and Retention

Frank E. Ritter, Gordon Baxter, Jong W. Kim, and Sowmyalatha Srinivasmurthy

Date: 1 sug 2010/ Due 1 aug 2010

(For submission to the *Handbook of Cognitive Engineering*)

Manuscript Statistics

Word count: 7,723 excl. refs but including figures (target 7-8k words)
Figures: 6
Tables: 0

Corresponding Author

Frank E. Ritter (frank.ritter@psu.edu)
College of Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16802

Abstract

Learning, the rate it occurs, and the stages learning goes through are important aspects of human behavior and should inform design more often. Learning provides performance improvements, sometimes drastic improvement with practice. This improvement can make designs that are too slow on first use to become more usable and remove errors in performance. The retention curve is similar, in that drastic skill loss is also possible and more dangerous. It too must inform design more often.

Keywords: Learning, retention, learning curve, transfer, skilled behavior

Table of Contents

1. Introduction

People get faster and make less noticeable errors as they repeatedly perform the same task. This is what most mean when they refer to *learning*, More formally, here we adopt Kimble's (1961) description of learning as the relatively permanent change in response potentiality that takes place as a result of extended practice.

By following a human-centered approach to design, we explicitly acknowledge that we are dealing with a population that encompasses users with a range of skills and abilities. We know that these skills and abilities will change over time as a result of learning, so we should explicitly support the learning process when designing systems. To do so we need to understand how learning takes place, and how it improves performance, sometimes dramatically, through practice. Systems that initially appear slow and effortful to use, often become quicker and more usable with practice—learning is therefore something that users and designers can exploit to compensate for particular design decisions.

In this chapter we discuss the relationship between learning and skilled behavior, noting the sorts of performance improvements that occur over time, and the process that underpins these improvements. The performance improvements can be modeled using learning curves. There are ongoing debates about the curve's shape with corresponding debates about the learning mechanisms and the implications for cognition and the underlying cognitive architecture. There can be a negative impact of practice on performance too, and performance can degrade over time through lack of continued practice. We describe how this relationship can be modeled using retention curves, before going on to consider the implications of learning and retention for system design.

2. Learning and Skilled Behavior

Performance changes with practice, typically getting faster, less effortful, and exhibiting fewer noticeable errors. These changes apply to most tasks that people carry out, including the use of computer-based systems. As users perform tasks, they acquire new information and skills and, through practice, improve the strength of their memories. More frequent retrieval of information or learnt skills may lead to the improvement of memory strength too: retrieving a password is faster and more robust after 2 attempts than 1, and after 100 attempts than 99 (all else being equal), although the increase in performance diminishes with successive attempts.

Human performance can be described using Neisser's (1976) Perceive-Decide-Act cycle and similar frameworks (e.g., Coram, 2002; Newell, 1990). People perceive the current situation, which leads them to make decisions about which actions to take, and then perceive the updated situation, and so on. Elements in the task environment are initially perceived individually, but related elements become grouped together with repeated exposure. Perception of the situation hence becomes more holistic with practice, and recognition consequently becomes quicker. People also come to associate decisions and responses with these learned patterns. In the same way, users can learn to adjust their behavior to accommodate the idiosyncrasies of a particular interface: where it is not immediately obvious which actions can be performed, users typically must learn which actions are allowed. Similarly, users often have some predetermined beliefs about where particular objects should be on the screen, and if they do not appear where they are expected, their location must be learned. These predetermined beliefs can be called a mental model (Moray, 1999), and represent a theory of how the world is structured and works. If interfaces are appropriately designed, learning can happen without causing the user much distress; poorly designed interfaces are usually not appreciated because considerable time and effort has to be expended on learning.

We will present a summary, explanatory theory of how users learn, building on existing theories. Our theory has many implications, including how user's behavior changes with learning, what extreme practice looks like, and suggestions for system design. While we tailor our discussions towards learning in the use of computer-based systems, there are many more texts on psychology and cognitive science that provide more general discussions on human learning (e.g., Anderson 1982; 1995; 2004; Lindsay and Norman, 1977).

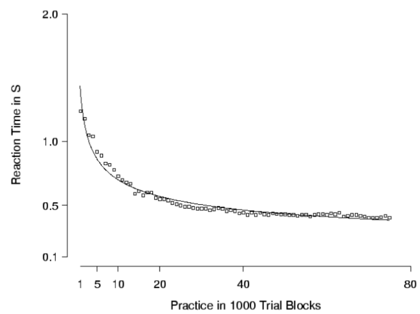
(1995; , 2004; , 1981; Lindsay & Norman, 1977)

2.1 Improvements from Learning

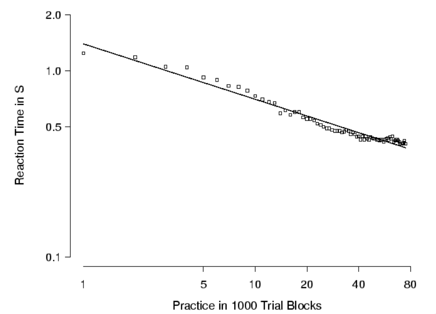
Perhaps the biggest regularity of human behavior in general, and particularly user behavior, is changes due to learning, which lead them to get faster at performing a task the more they do it. In most tasks where it has been studied, the relationship between the number of trials and performance time follows similar shaped curves. The range of tasks is very wide, covering pushing buttons, reading strange text, doing arithmetic, typing, using a computer, generating factory schedules, all the way up to writing books (Crossman, 1959; Ericsson, Charness, Feltovich, & Hoffman, 2006; Nerb, Krems, & Ritter, 1993; Newell & Rosenbloom, 1981; Ohlsson, 1992; Seibel, 1963). These learning curves also apply when people work together in groups (e.g., Lieberman, 1984; Wright, 1936).

There are large improvements initially, although users rarely report satisfaction with these improvements. These improvements follow a monotonically decreasing curve over time. In the Seibel task (1963), participants are shown a pattern of 10 lights and have to push buttons corresponding to those lights that are illuminated. Each point on the curve for this task (see Figure 1) represents the average reaction times for doing 1,000 patterns. With practice behavior improves, but by a reducing amount. Over a wide variety of tasks and out to hundreds of thousands of trials, people continue to get get faster (e.g., Crossman, 1959; Seibel, 1963).

With such large time scales it becomes difficult to see the incremental improvements in performance time. Using logarithmic axes (see Figure 1 (b)), makes it easier to perceive the incremental changes. Typically, on these log-log plots, learning curves more or less follow a straight line.



(a)



(b)

Figure 1. Time to perform a simple task (pushing a combination of buttons) (a) on a linear plot, and (b) on a log-log plot. The solid line on each plot shows a power law fit to the data. (Adapted from Seibel, 1963, and previously used in Ritter and Schooler, 2001).

The mathematical representation of the learning curve is currently disputed. While most learning curves generally follow a straight line in log-log plots, there are some systematic discrepancies. Some researchers believe that the learning curve represents an exponential function, while others think that it is a power equation (thus, called the power law of learning) of the following form:

$$\text{Time of a trial} = \text{Constant1} + \text{Constant2} * e\text{^}(\text{Number of trials} + \text{PP}) \qquad \textbf{Equation (1)}$$

$$\text{Time of a trial} = \text{Constant1} + \text{Constant2} * (\text{Number of trial} + \text{PP})\text{^-}\ \alpha \qquad \textbf{Equation (2)}$$

[Note to typesetter: rewrite these equations using an equation tool]

Constant1 is the minimum time based on the machinery or external environment. It is sometimes omitted because it makes the regression harder to compute. It can be computed from equations describing the world or by using physical equipment. For example, you might measure how long it takes a ball to fall to the ground with a camera if the task involved catching a falling ball, or you might record how fast an interface can accept keystrokes when driven by a computer program. *Constant2* is the time that decreases with practice; it is always used.

PP is previous practice on the task. *PP* is either estimated, measured directly, or ignored. It is usually ignored because the task is unique enough to the learner. In other cases, such as taking up a familiar task, the equation does not fit as well if it is omitted.

α is typically in the range 0.1-0.5 (reviewed in Newell, 1990, Ch. 1; also see Anderson, 1995, Ch. 6, and Heathcote et al., 2000 for example values). α is usually found by plotting the data using log-log axes, and fitting a straight line using a linear equation (with a known closed form solution) to the log of the trials and the log of the time. The more accurate, but more computationally expensive approach, is to fit the power law equation, including the appropriate constants, using an iterative algorithm.

Being able to mathematically predict learning is important for several reasons. For science, it helps summarize learning, and comparing the constants (*Constant1,* *Constant2*) provides a useful way of characterizing tasks. It also has practical applications. In engineering, design, and manufacturing, it can be used to predict how fast users will become with practice, and hence can be used to predict factory output and profitability (e.g., Lieberman, 1984).

An alternative interpretation of the way improvements occur through learning suggests that the match to a power law is an artifact of averaging the data from multiple people and multiple series. The curve is therefore best described as an exponential when the data is examined in its purest form (Heathcote, Brown, & Mewhort, 2000). Both interpretations have basically the same implications for users with limited lifetimes and physical equipment. (They have, however, different implications for the details of learning and its relation to the mechanisms of cognition and knowledge.)

In addition to reducing performance time, several other aspects of behavior improve with practice too (Rabbit & Banerji, 1989). The number of noticeable errors decreases, usually because people detect and correct the errors before their consequences become apparent to casual observers. The variance in task time also decreases (Crossman, 1959; Rabbit & Banerji, 1989). Some believe that this decrease in variability is the greatest contributor to improvements in speed, because the minimum time to perform a task generally remains unchanged with practice. There are notable exceptions, however, such as those tasks that cannot be completed by novices.

There appear to be two instances where learning cannot get any faster. Users cannot get faster when the machinery they are working with cannot keep up with them. This was first noted when cigar rollers improved up to a point and then stopped. It was found that the users were rolling faster than the machine could pass them materials (Crossman, 1959). This constraint is represented by *Constant1* in the equations above.

The second instance is when they change strategies. As a user adopts a new strategy, they often experience a slowdown, moving back on the practice curve for that strategy. With further practice, performance on this curve with a lower intercept improves, usually to be much better than the previous strategy (Delaney, Reder, Staszewski, & Ritter, 1998) (see Figure 2).

If your system allows multiple strategies, you should consider supporting user transitions between them. In text editors, for example, there are several ways to find a target string, including scrolling line-by-line, scrolling by paragraph, and searching. In one study (Card, Moran, & Newell, 1983), most users employed a sub-optimal strategy (scrolling line-by-line) rather than searching; experts typically use searching, which can be 100 times faster than scrolling.

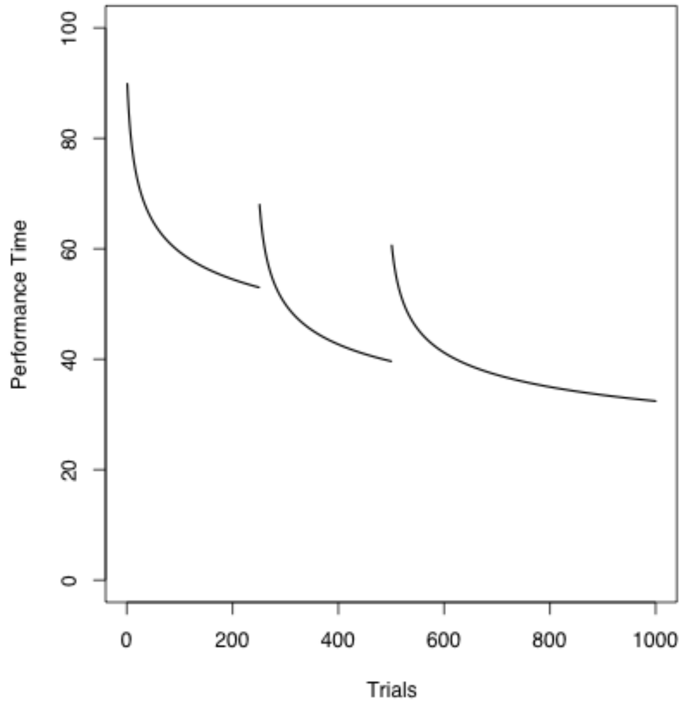


Figure 2. The learning curve for a series of better strategies, showing the initial slow down and then savings due to switching strategies. (Diagram by Ritter)

2.2 Theories of Learning

Theories that describe the general process of learning have been developed in several areas, including behavioral psychology (Fitts, 1964), cognitive psychology (Anderson, 1982), and cognitive engineering (Rasmussen, 1983). These theories all propose that learning occurs in a number of stages. We will focus our discussion mostly on Rasmussen's approach (see Figure 3), whilst noting the parallels with the theories of both Fitts and Anderson.

2.2.1 Stage 1: Declarative knowledge acquisition

The learner first acquires declarative knowledge about the domain. Problem solving is difficult at this stage. The domain concepts may not always appear connected to each other. The learning might be considered as learning how to learn more about the task. What is learned at this point, such as the initial problem solving state and the actions (physical and mental) available for problem solving, may not be enough to complete the task. Problem solving, when it is possible, is effortful, and not always correct. The user might not have much confidence in their performance. Rasmussen describes behavior at this level as knowledge-based; Anderson calls this the cognitive stage.

When learners are at this first stage, behavior occurs at the most fundamental level. There are no direct rules to inform the user what to do next, so they usually have to work from first principles, using strategies such as trial and error. Expert users can find themselves at this level in emergencies or novel situations, when deliberate thought and reasoning about the state of the situation or system are required, based on the user's knowledge (i.e., mental model) of the system. Getting your car out of a deep rut, or navigating in an unfamiliar town are examples from driving. In computer systems, installing software that does not want to install, and changing UNIX permissions on shared files would be other examples.

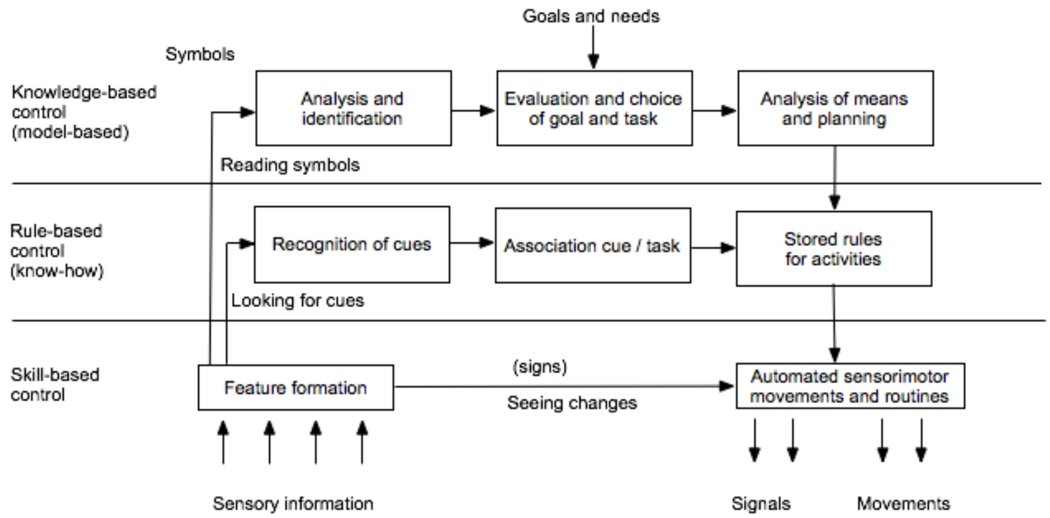


Figure 3. A graphical representation of Rasmussen's theory of knowledge levels (Redrawn from Rasmussen, 1983).

2.2.2 Stage 2: Proceduralization

With learning and practice, the user progresses to Rasmussen's rule-based behavior (what Anderson calls the associative stage). In this stage the learner can solve problems more routinely and with less effort by following a sequence of steps, or procedure. The declarative knowledge has been compiled into procedures that are relevant to the task domain and can be performed directly. Users may often be able to recognize what needs to be done and apply the appropriate procedure. For users of complex systems, behavior at this level is a consciously controlled activity and is based on familiar rules, either dictated or acquired. Lane changing in a car would be an example, as would docking a ship in an infrequently visited port. Use of office application software similarly often requires rule-based behavior to insert a figure into a document, or format a paragraph.

2.2.3 Stage 3: Skill tuning

The final stage, where the knowledge that is applied is refined gives rise to skill-based behavior (Rasmussen); Anderson calls this the autonomous stage. At this point users still get faster at the task, but the improvements get diminishingly smaller. New declarative information is rarely learned, but small adjustments to rule priorities happen. At this point users are generally considered to be experts. Behavior is much more automatic, and occurs when users perform routine functions in their normal operating environment. Much of their behavior is no longer consciously controlled, and hence cannot be verbalized because the behavior is unavailable to conscious thought. Users perform the task faster and, because little or no conscious control is required, this frees up attentional resources that can be allocated to other tasks. Simple lane following by an experienced motorist would be an example. Pressing an on button for a computer or deleting text in a word processor might be examples in HCI.

2.2.4 Moving between the stages

The improvements in performance time do not appear to delineate the stages of learning noted above, except in cases where individual data are studied and strategies are known to take different times (e.g., Delaney, Reder, Staszewski, & Ritter, 1998). This may be because the first stage of learning, where performance might be quite difficult, has to be nearly complete before the task can even be performed. There are hints of this in Figure 1 (b), where the initial slope is perhaps more shallow than would be expected. In complex tasks, the transition of rule learning and rule tuning within tasks might lead to steady improvement, and be masked by the large number of rules and sub-tasks. Looking at individual users working on well measured tasks may allow stages to be seen. When this has been done, strategy changes can be identified (Delaney, Reder, Staszewski, & Ritter, 1998).

2.2.4 The stages in practice

These three stages of learning have been noticed in several areas of formal reasoning, including physics problem solving, where the link between problem solving and learning has been studied (Larkin, 1981; Larkin, McDermott, Simon, & Simon, 1980a, 1980b). Formal reasoning is problem solving with known goals (like solve for *x*), using a closed set of equations or rules that can transform representations (like adding 2 to each side of an equation). Examples include proofs in geometry and solving algebra problems.

In these types of problems, the problem solvers (and computer models) start with domain knowledge and the goals they are trying to achieve. Once novices have learned declarative knowledge, they can then work backward from the goal. In the case of physics problem solving, the task is to derive a value of a variable (like *final speed*), given some known variables (like *mass, initial speed*, and *force*), and some equations (like *Force* = *Mass* × *acceleration*). Novices tend to work back from the quantity they are trying to find (*final speed*), chaining inference rules together until they find known variables. If they need to compute the *final speed*, they look for an equation that computes *final speed*. Then they look for more equations to find the variables in the first equation, until they bottom out with known variables. This is known as backward chaining, or bottom-up reasoning. After each successful application, a new operation is learned that supports applying that formula to find a variable. After enough learning, experts start with the unknowns, find or recall equations that apply, and work towards the goal in a forward-chaining or top-down way.

Similar learning strategies will be seen in computer repair and troubleshooting, software usage, and other domains where formal reasoning can be applied. Better interfaces will support the novice by providing the domain knowledge needed to learn the inferences, and will support novices and experts by providing state information to reason from.

Implicit learning can occur when the user is working at the skill-based level, with knowledge eventually being derived at the knowledge-based level through reflection on one's own behavior. This type of learning, which often leads to, or arises from strategy changes, is an active area of research in psychology (e.g. French & Cleeremans, 2002).

Whichever way the user learns, some key phenomena persist:

- The ability to recognize correct / incorrect items comes before the ability to generate correct items.
- Knowledge is not acquired in an all-or-nothing way. Novices go through a stage of fragile knowledge, where sometimes the correct knowledge is used, and sometimes not.
- Experts acquire a rich repertoire of representations of their knowledge. Experts not only know more than novices, their knowledge is much better organized and more readily available.

2.3 Alternative Views on Learning

There are several other ways of describing learning. One common way is to distinguish between declarative and procedural types of learning. Another common way is as implicit and explicit. The distinctions between these classifications is still subject to debate, but they represent useful and interesting differences about users that can inform system design.

Declarative learning is learning facts (declarations), such as "the power button is on the keyboard", and "the computer manager's office is 004A". There are two types of declarative memories: recognition and recall. Recognition memories are easier to build than recall memories. That is why multiple choice tests seem easier—you just have to recognize the answer.

Procedural learning is learning acts, or how to carry out procedures. Typing, using an interface, and playing a computer game are all examples of this. Procedural memories are more complex than declarative memories in that they generally support the skill being performed in a wide variety of contexts, which thus represents more knowledge. These memories are formed after the declarative representations are available to create them.

These two types of learning are characterized by different regularities. Declarative learning can, by definition, be directly described and reported; procedural memories cannot (Ericsson & Simon, 1993). You cannot directly describe the procedural knowledge you use to ride a bike, although you can accurately report the declarative knowledge you used to generate the procedures (like keep your weight balanced), and what is in your working memory as you perform the task (there is a parked car ahead). You can also watch yourself do a task and attempt to describe what you were paying attention to as you did it. What you think you were paying attention to, however, depends on your mental model of the task and how demanding that task is: this approach is called introspection.

There are fundamental problems with introspecting in this way. While introspection can lead to useful and helpful insights, it does not lead to complete and valid theories of human thinking (Ericsson & Simon, 1993). Mainstream psychology has rejected introspection as an accurate measure of how people think because it is subject to many biases and does not have access to most mental processes, although one may find introspection a useful inspiration for ideas that can be later validated by other approaches.

The problems with introspection can be illustrated by a study that looked at learning key bindings in a text editor. When key bindings in the editor were changed on the second day of the study, for the first hour or two the users felt much, much slower. They were slower than they were at the end of the first day, but still faster than when they started. They regained their skill level by the end of the second day (Singley & Anderson, 1989). This study illustrates three important things about users. The first is that they are always learning. The second is that introspection often leads to incorrect conclusions: they rather disliked the new interface, but adapted more quickly than they thought they did. The third is that the users were able to transfer or relearn all of what they had learned from the old interface to the new interface, at least as measured by task times. Only the key bindings changed; the underlying approach and the command structures did not, so the users were still able to apply most of what they had learned. Another similar example found that users preferred mice over light pens, even though the latter were faster to use (Charness, Holley, Feddon, & Jastrzembski, 2004).

In the implicit/explicit distinction, implicit learning seems to be automatic, is based on practice, is not improved by reflection, and produces knowledge that cannot be verbalized. This bears some similarity to the rule tuning stage. If the rules are created based on a simple theory of the domain, but a more complex theory of the domain exists, then additional learning can occur.

Explicit learning proceeds with full consciousness in a hypothesis testing way. It produces knowledge that can be verbalized.

These distinctions become important when teaching users how to use an interface. Some information is presented to them as declarative knowledge to be learned (where things are, who other users are, and what the objects are), and some is presented as procedural skills, such as how to do a task. If the declarative information is not available, the learning will be implicit.

Learning can also be massed or distributed. Massed learning occurs at a single time, for example, cramming for a test. Distributed learning occurs with breaks in time between the learning episodes. Figure 4 shows how much better distributed learning can be for the same amount of practice. This figure is based on the ACT-R declarative memory equations. Distributed learning takes less total time (sometimes 1/3 of the time), and the retention is better, sometimes 50% better. Distributed learning also allows more time for learning and may lead more time spent learning. Anything you can do to help your users learn in a distributed fashion will facilitate their retention. Some interfaces now put up hints which appear to be a way to support distributed learning.

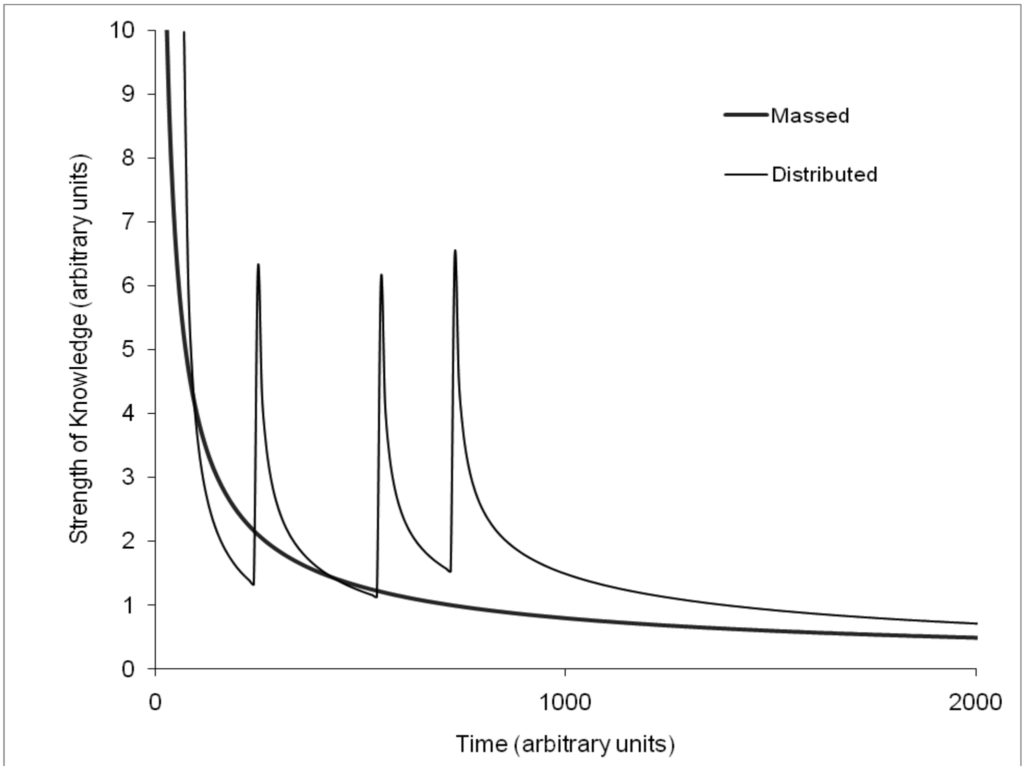


Figure 4. Massed vs. distributed practice in relation to time to learn a list. Given exponential decay, which is often assumed for memory, the solid line shows a given amount of practice as one block, and the dashed line shows the same amount spread out over a longer period. The distributed practice has higher activation after the third practice, and will for the remainder of the curve.

Finally, we should note again that learning is rarely complete. It is easy to think that if you have learnt to operate a system your learning about that system is complete. This need not be so, many users can sometimes perform very competently with very little knowledge. Other complex systems, such as operating systems, some computer applications, and vehicles of all kinds are complex enough, that once competent, users can continue to learn new ways to use them and new components for years.

2.4 Skilled Behavior, Users in Complex Environments

Human performance is often a complex mixture of behavior at different levels. In most cases, routine users will exhibit open-loop behavior moving seamlessly between the rule-based and skilled levels. That is, they will be able to perform most tasks in a routine way using existing, well-used knowledge without checking all of their steps. They will not close the loop by checking that things worked correctly because in most cases they no longer need to. If they do make errors, they will be able to recognize and correct them quickly, often before any adverse consequences occur. They will continue to get faster with practice, but the improvement will be minor. These skills will be easier to compose with other tasks. There will also be some closed-loop behaviors for tasks that are less well practiced, and on these tasks users will be more careful and check their work.

Aircraft pilots, for example, often operate at all three levels. Some aspects of their behavior are automatic, for others they refer to rules and procedures, whilst for others, particularly in emergencies, they reason on the basis of their knowledge about the plane and learn. For most emergencies, they will use the quick reference handbook (QRH), which provides them with compiled knowledge in the form of checklists. In unusual emergencies, however, which are not documented in the QRH, they have to reason from first principles.

Rasmussen (1983) argues that good design needs to support all three levels of behavior, not just one. We can also note a social human factor here: if there are multiple team members, they may operate at different levels at the same time due to differences in experience. This may give rise to conflict or consensus, depending on how these differences are resolved. In an emergency situation in an aircraft, for example, the more experienced pilot (the Captain) may be working at one level of reasoning, while the more junior pilot (the First Officer) may be using a different level that may not be appropriate to the unusual situation (or vice versa). Crew Resource Management (Wiener, Kranki, & Helmreich, 1993) is an approach to reduce this problem.

The SwissAir crash at Halifax, Novia Scotia in 1998 illustrates the point about pilots operating at different levels (e.g., see Carley, 1999, January 21; Dekker, 2005). After a burning smell was detected, there were reports of smoke in the cockpit. The first officer, who was flying the aircraft, suggested that they should immediately start jettisoning fuel to make the aircraft light enough to descend and land at the earliest opportunity. This instinctive behaviour is typical of the Skill-Based level. The captain, however, decided that they should first follow the predefined procedures for dealing with smoke and fire. He was reasoning at the rule-based level. The decision about dumping fuel was delayed. Meanwhile the fire continued to develop, and the aircraft became uncontrollable, and eventually crashed into the sea, killing all 229 people on board. .

2.5 Expertise and Experts

With extended practice users become experts at their task. Generally, to become world class, it takes about 10 years of practice as well as some deliberate reflection and declarative learning, either from extensive self-tutoring or a coach (Ericsson, 1996; Hayes, 1981, ch. 10; Schneider, 1985; Simonton, 1996). There are exceptions, but most authors argue that these are rare or nonexistent. Less time is required for local or national prominence. And deliberate practice is a usual and necessary, but not sufficient condition for expertise to develop (Ericsson, 1996; Hoffman, Shadbolt, Burton, & Klein, 1995).

Deliberate, intensive practice with a tutoring system can provide an usually large amount of time on the critical aspects of a task, where field users have to spend additional time to see the critical aspects, learning can be greatly accelerated. For example, the Sherlock tutor (Lesgold, 2001) abstracts the physical movements to learn a diagnostic reasoning task. Students using the tutor for approximately 25 hours learn as much as others learn after 4 years of practice in the field. Learners in the field have other tasks, but also the steps in the diagnostic sequence can take hours to perform.

There are some theories that suggest that with practice the user gains more than just speed in the task. They appear, in some tasks and in some studies to have greater memory and more attention that they can pay to the task (Ericsson & Kintsch, 1995). In nearly all cases they have more accurate perception for and of the task details. In all cases, they have more knowledge and better anticipation of what will happen in the task.

2.6 Transfer

One of the important issues in learning is transfer of learning. After a skill has been learned, the goal is often to reuse the knowledge or apply it to a new situation. If no transfer occurred, every situation would be a new situation; if transfer was perfect, few situations would be novel. There are both positive and negative examples of transfer: in some cases transfer does not appear to happen very effectively; in other cases, it can be quite effective.

A telling study on transfer was done by Gick and Holyoak (1980) based on work by Duncker (1945), where, before solving a problem, subjects read about an analogous problem. The story (analogous problem) was about how a king wished to invade a town, but his army was too big to come through a single gate in the town, so he split his troops into smaller parties and entered the town through several gates. Subjects then read about the use of rays to attack cancer: the problem was that the beam of rays was too intense to be directed at the cancer cells without seriously damaging the surrounding tissue. A surprising number did not think to use multiple beams of rays, which is what the transfer of knowledge would suggest. This lack of obvious (to the experimenter) transfer effect has been repeated many times. Thus, users do not always transfer useful information unless the information is very well practiced, the transfer is very straightforward, and they have time to think.

A counter example, however, is provided by the previously noted study on the use of perverse Emacs (Singley & Anderson, 1989). They found that transfer can occur, but users do not always see it, and they do not like it when transfer leads to small mistakes (as would have happened often in the first hour for the users of perverse Emacs). Thus, changing key bindings is likely to lead to frustration, but might not hurt users in the long run because the deeper knowledge would transfer.

When complex concepts and tasks need to be learned, they may be introduced using simple analogies: the way muscle fibers work, for example, can be likened to the way that a rowing crew operates (e.g., Spiro, Feltovich, Coulson, & Anderson, 1989). The idea is that learning can be transferred between the simpler concept or task, and the complex concept or task. Whilst this can be a useful approach initially, the simple analogies can lead to misconceptions if they are inappropriately applied to situations where they do not properly reflect the real situation. To overcome these limitations, multiple analogies need to be used, an approach that is advocated by cognitive flexibility theory (e.g., see Feltovich, Spiro, & Coulson, 1993). So for the muscle fiber example, you also have to introduce analogies that overcome the limitations of the other analogies, and end up with a set of five analogies that together help to capture the full complexity of how muscle fibers really work.

There are tools available to measure potential transfer. One example is GOMS (Bovair, Kieras, & Polson, 1990; Kieras, 1998), a simple language to express how to do a task. Each instruction in the language, such as how to push a button, takes about 17-30 s to learn,) while declarative pieces of knowledge take about 6-10 s. Thus, the amount of differences between how to do two tasks indicates how much knowledge can be transferred and how much must be learned. Analyses of popular operating systems suggest that there are real differences between them, with one being a subset of the other, and one thus being easier to learn (Kieras, 1998).

3. Retention

Pilots have to go back to training school on a flight simulator every several months, and they have to hand fly at regular intervals to maintain their accreditation to fly particular types of aircraft. Many professionals are in this situation. In this section, we discuss this maintenance, or retention of knowledge and skills. Schmidt and Bjork (1992) argue that learning is an imperfect indicator of later performance, and that learning and retention, therefore, should be considered together. It is therefore important to understand how acquired knowledge and skills are retained and should be maintained in the longer term.

3.1 The Process of Retention

Earlier, we discussed the process of learning—there is a consensus understanding of how learning occurs in three different stages—which is supported theoretically by the ACT-R cognitive architecture. Based on this theoretical foundation, we can describe how forgetting and retention might vary reliably across those corresponding stages of learning. Figure 5 provides a summary of learning and retention accounts.

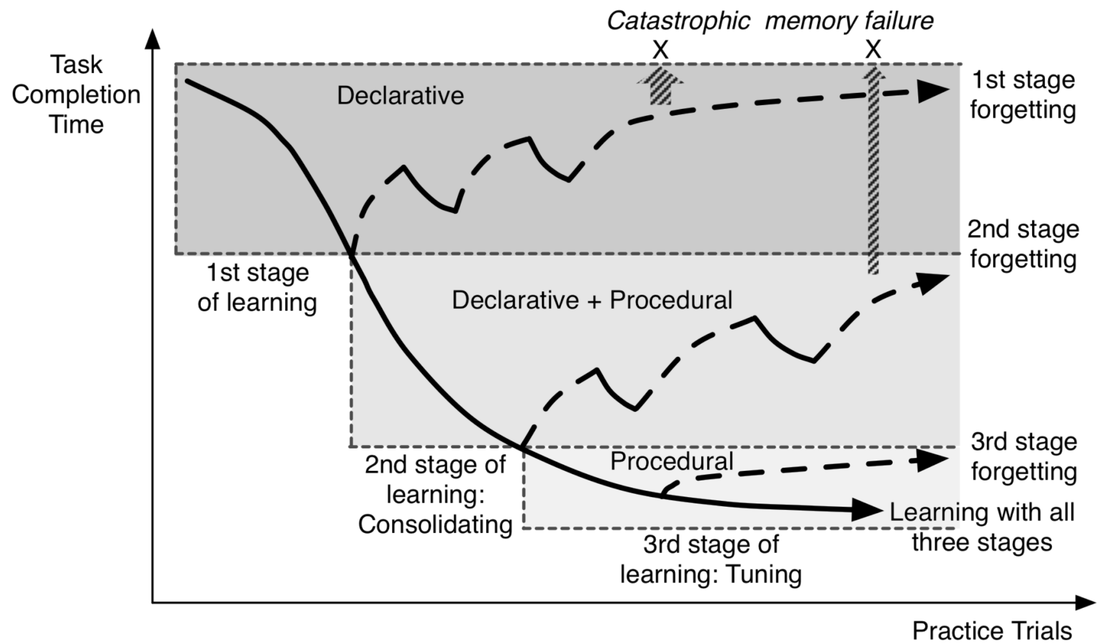


Figure 5. A theory of skill retention, showing the three stages of learning and forgetting. Rasmussen called these stages knowledge-based, rule-based, and skill-based. Fitts called these stages cognitive, associative, and autonomous.

The First Stage: Declarative. For this first stage of learning and forgetting, knowledge in declarative memory degrades with lack of use, perhaps catastrophically as indicated by X's in Figure 5, leading to the inability to perform the task. In this stage, learning and forgetting are accounted for by the activation mechanism for declarative knowledge in ACT-R: with lack of use, the strength of declarative memory items declines. Decreased memory strength leads to increased response times and decreased retention and performance accuracy.

The Second Stage: Associative. In the second stage of learning, task knowledge is represented using a mix of declarative and procedural memory. With lack of use, the declarative knowledge is forgotten, leading to mistakes and missed steps. Procedural memory, on the other hand, is basically immune to decay. The slope of the forgetting curve in this stage could vary by subtask because it would vary in their knowledge mix, and different mixes would decay at different rates. In the first two stages, catastrophic memory failure can occur because the declarative knowledge is not fully activated. This finding suggests that in this mixed stage training is necessary to keep the declarative knowledge active and also to support further proceduralization because the activation of declarative memory is required to generate procedural rules.

The Third Stage: Procedural. In the third stage of learning, task knowledge is available in both declarative and procedural forms, but procedural knowledge predominantly drives performance. Practice will compile declarative knowledge into procedural knowledge. We describe this type of task knowledge as a *proceduralized skill*. With lack of use, declarative knowledge may degrade. Nevertheless, the learner can still perform the task if all the knowledge is proceduralized and thus not forgotten with time, and performing the task does not require new declarative inputs.

4. Conclusion: Implications for System Design

4.1 Learning and Design

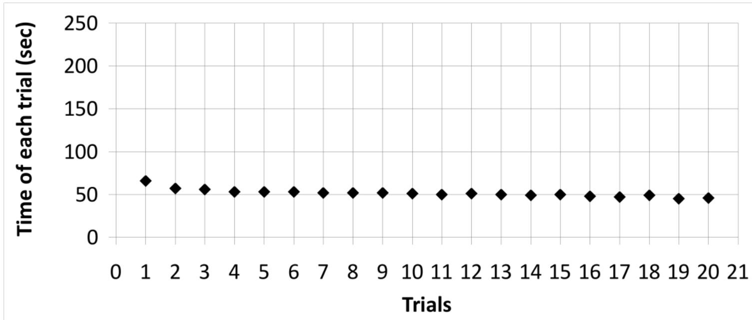
The theories and results noted here have several implications for system design. It is important to consider the different levels of behavior when designing systems. You need to support users learning the basic skills at the knowledge level and as they progress to rule-based and skill-based behavior, which are both more efficient. The support needs to include the provision of appropriate information that will help them operate on the rule-based and skill-based levels.

All users will learn and hence get faster at doing a task with repeated practice. This is an important way that users fit themselves to a task, covering up errors in design or compensating for trade-offs made in interface design that do not favor the user. Interfaces that are initially too slow may become faster, but will require training (to learn the task) and practice (to get faster). These times can be roughly predicted by comparison with other tasks or after several initial trials to get the slope of the improvement. There are theories that predict this, based on a simple task analysis (Kieras & Bovair, 1986) where a step (rule) takes about 30 s to learn. Work continues on more complete theories both from a design perspective (Christou, Ritter, & Jacob, 2009) and a cognitive architecture perspective (Anderson, 2007).

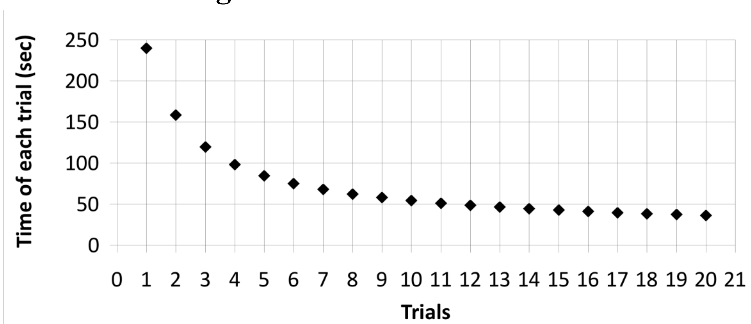
Users must have some declarative information about a task before they can perform it. This means that interfaces and their associated system, including manuals, online help, web sites, and other users, should provide new users with enough information to perform the task, or at least to start to explore the environment.

Both the procedural and declarative memory usage depends on how they are represented. If the user has a more general representation and declarative knowledge associated with it, the procedural knowledge can transfer to new problems more easily. Interfaces and systems (including instructional materials) that support more general representations and more task-oriented representations (vs. tool specific representations) will allow users to transfer their knowledge both in and out of the interface more readily.

The learning curve that can be created by observing learners can provide insights to system design directly. The start of the curve shows how difficult the interface is to novices. This amount of time might be seen to be too high, if it is, or at an acceptable level. Figure 6 (a) shows a relatively shallow curve that shows that novices are not so different in performance than experts. Figure 6 (b) shows a curve where novices would be less happy, but which might be appropriate for a game. The slope of the curve can also provide insights. If the curve is very steep, and the system will be often used, the initial task times might not be an issue. If the interface is used only a few times, then the slope needs to get the user’s task time to an acceptable level more quickly. It can be the case that 150 s is an acceptable time, or it can be that 50 s is not and even the well learned portion of the curve is not yet fast enough—it will depend on the context. (For an example of a task with thresholds, see Bryan & Harter, 1897.)



(a) A relative shallow learning curve.



(b) A relatively steep learning curve.

Figure 6. Two learning curves with approximately the same final speed. (a) shows a shallow learning curve and (b) a steeper and longer curve. A relatively shallow curve may be more appropriate for interfaces not used often, while a relatively steep curve may be more appropriate for interfaces used by experts.

Some users like to learn. This is probably most often seen in games, which have their explicit goal to teach the user some procedural skill (like driving a racing car), or some declarative knowledge, like the way around a dungeon. Many other users prefer not to learn, but will if the task is important enough. This is likely related to a trait called need for cognition (Cacioppo & Petty, 1982), which notes that some people like to think and learn more than others.

Systems that allow users to recognize actions they want to do (e.g., by using menus) will be easier initially to use than those that require users to recall keystroke based commands. There is a tradeoff, however, when the novices become experts. The experts will be able to recall the keystrokes or command names and will not wish to wade through the choices. For example, Linux experts like to use keystrokes in the command shell, while novices on Windows prefer a graphical user interface.

Systems that encourage users to reflect on how to perform the task may lead to different types of learning. Where the learning is incidental and not tied to educational intent or systems, this may not matter. On the other hand, tutoring systems should be careful that what users learn is what is intended. For example, the interface should not encourage users to just try key presses or button clicks to see what happens, which can lead to implicit learning that is tied to the representation of the interface and not to the domain.

4.2 Retention and System Design

In various systems (or interfaces), the process of learning occurs as human users interact with those systems. Wherever learning is considered to be important, retention should be also taken into consideration. For example, training systems is an important domain in which improving the user's performance and skill retention are the goal of the system.

We can analyze a training system in terms of the aforementioned three stages of learning and retention. This approach provides a way to evaluate the effectiveness of the training system—a cognitively engineered training system can optimize the scheduling of necessary training (e.g., Atkinson, 1972; Pavlik, 2007) by helping the trainee to achieve long-term retention. We present potential principles for a cognitively engineered training system.

When it comes to the first stage of learning and retention, the trainee's knowledge in declarative memory would degrade with lack of use, leading to the inability to perform the task. With lack of use, the strength of declarative memory items declines, following an activation equation. The base-level activation taken from ACT-R, and shown in Equation 3 illustrates this relationship for a chunk *i*. B_i , the strength of a memory is related to β, a constant, n is the number of presentations for a chunk i, t_j is the time since the j^{th} presentation, and d indicates the decay parameter.

$$B_i = \beta + \ln\left(\sum_{j=1}^{n} t_j^{-d}\right) \qquad \text{Equation 3}$$

In the second stage of learning, training is needed to avoid catastrophic failures in task performance resulting from the learner's inability to retrieve declarative knowledge for steps still requiring declarative knowledge elements, either because the task is incompletely proceduralized or because declarative inputs vary for the task (e.g., the radio frequencies or flight numbers used by pilots as they enter or exit controlled airspace). The higher decay rates associated with declarative memories also suggest that achieving more complete expertise entails training not only common but also uncommon tasks to the procedural level (e.g., emergency procedures for pilots).

Furthermore, while the periods of time between retraining sessions can increase with practice, catastrophic failures can still occur if completing the task requires declarative memory elements (i.e., changing inputs like frequencies, coordinates, or names). As an example, in learning typing skills, this stage would correspond with the learner possessing a strong declarative representation of the key locations while also possessing some procedural rules regarding the task as retrievals increasingly get combined with key presses (in other words the greater association of conditions with actions).

The third stage of learning and retention would correspond to the knowledge about how to type be fully represented in production rules, and the declarative knowledge would not get used any more and would decay. Infrequently used characters (e.g., # or ^), however, might still require declarative retrievals if these characters are insufficiently practiced.

Conversely, less practiced or infrequently used skills, such as responding to unusual errors, may exhibit the mixed curve shown in Figure 5. Consequently, these skills require concerted and structured practice to proceduralize. Unlike the common ASCII keys in the typing task, there is no assurance that routine task execution will compile and proceduralize the declarative memory elements associated with the task, meaning that training is most likely necessary to achieve proceduralization (noted in Figure 5 as crossing the dashed grids that represent the stage thresholds).

4.3 Learning and Higher Order Cognition

Learning also can be used to help with higher order cognition constructs such as attention and situation awareness (SA, consider cite to chapter in handbook). With greater learning people begin to recognize situations and generate responses faster. This allows them to allocate spare attentional resources to a complex task or to a secondary task.

Learning has a secondary effect on SA by supporting many of the activities related to SA and generally improving SA. Greater learning means that each step from perception to action will be faster and more accurate. Faster processing times mean more time is available for monitoring the situation, and less working memory decay while performing secondary tasks. Greater learning means more knowledge is available to predict the world and that these predictions will be more accurate.

4.4 Summary and Further Reading [Editor: put in cross cites]

Learning of knowledge and skills and subsequent retention of these skills are fundamental aspects of users. There are many known regularities in learning and retention. The ones reviewed here provide a short introduction. There are more topics and factors in learning and retention that are worth understanding, including chunking, compilation of knowledge, conscious control, expertise, learning from reflection and self-explanation, scaffolding, and learning from errors. All of these regularities have implications for design that are both feasible and powerful.

4.5 Further Outside Readings

We can recommend the following further readings in learning and retention in addition to the cited works.

Anderson, J. R. (1995). *Learning and memory*. New York, NY: John Wiley and Sons.

This book provides an overview multiple theories and sources of data on learning.

Jonassen, D. H., & Grabowski, B. L. (1993). *Handbook of individual differences, learning, and instruction*. Hillsdale, NJ: Erlbaum.

This book provides chapters discussing the application of learning theory to education and explains many related topics.

Reigeluth, C. M. (2007). Order, first step to mastery: An introduction to sequencing in instructional design. In F. E. Ritter, J. Nerb, T. M. O'Shea & E. Lehtinen (Eds.), *In order to learn: How the sequences of topics affect learning* (pp. 19-40). New York, NY: Oxford.

This chapter introduces the ideas of instructional design, and highlights the strengths and weaknesses of several commonly used instructional designs.

References

[Ritter maintains]

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological Review, 89*, 369-406.

Anderson, J. R. (1995). *Learning and memory*. New York, NY: John Wiley and Sons.

Anderson, J. R. (2007). *How can the human mind exist in the physical universe?* New York, NY: Oxford University Press.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review, 111*(4), 1036-1060.

Anderson, J. R., Greeno, J. G., Kline, P. J., & Neves, D. M. (1981). Acquisition of problem-solving skill. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition*. Hillsdale, NJ: Erlbaum.

Atkinson, R. C. (1972). Optimizing the learning of second-language vocabulary. *Journal of Experimental Psychology, 96*, 124-129.

Bovair, S., Kieras, D. E., & Polson, P. G. (1990). The acquisition and performance of text-editing skill: A cognitive complexity analysis. *Human-Computer Interaction, 5*, 1-48.

Bryan, W. L., & Harter, N. (1897). Studies in the physiology and psychology of the telegraphic language. *Psychological Review, 4*, 27-53.

Cacioppo, J. T., & Petty, R. E. (1982). The need for cognition. *Journal of Personality and Social Psychology 42*(1), 116-131.

Card, S. K., Moran, T., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Erlbaum.

Carley, W. M. (1999, January 21). Airlines: Swissair pilots differed on how to avoid crash. *Wall Street Journal,* B1.

Charness, N., Holley, P., Feddon, J., & Jastrzembski, T. (2004). Light pen use and practice minimize age and hand performance differences in pointing tasks. *Human Factors, 46*(3), 373-384.

Christou, G., Ritter, F. E., & Jacob, R. J. K. (2009). Knowledge-based usability evaluation for reality-based interaction. In G. Christou, E. L.-C. Law, W. Green & K. Hornbaek (Eds.), *Challenges in the evaluation of usability and user experience in reality-based interaction* *(workshop proceedings). At CHI 2009 Conference on Human Factors in Computing Systems, Boston, MA, 2009. CHI 2009 Workshop: Challenges in Evaluating Usability and User Experience in Reality Based Interaction* (pp. 36-39). Toulouse, France: IRIT Press.

Coram, R. (2002). *Boyd: The fighter pilot who changed the art of war*. New York, NY: Back Bay Books/Little, Brown and Company.

Crossman, E. R. F. W. (1959). A theory of the acquisition of speed-skill. *Ergonomics, 2*, 153-166.

Dekker, S. (2005). *Ten questions about human error: A new view of human factors and system safety*. Mahwah, NJ: Lawrence Erlbaum.

Delaney, P. F., Reder, L. M., Staszewski, J. J., & Ritter, F. E. (1998). The strategy specific nature of improvement: The power law applies by strategy within task. *Psychological Science, 9*(1), 1-8.

Duncker, K. (1945). On problem solving. *Psychological Monographs, 58*(5), (Whole No. 270).

Ericsson, K. A. (Ed.). (1996). *The Road to excellence: The acquisition of expert performance in the arts and sciences*. Mahwah, NJ: Erlbaum.

Ericsson, K. A., Charness, N., Feltovich, P. J., & Hoffman, R. R. (Eds.). (2006). *The Cambridge handbook of expertise and expert performance*. Cambridge: Cambridge University Press.

Ericsson, K. A., & Kintsch, W. (1995). Long-term working memory. *Psychological Review, 102*, 211-245.

Ericsson, K. A., & Simon, H. A. (1993). *Protocol analysis: Verbal reports as data* (Vol. 2nd ed.). Cambridge, MA: MIT Press.

Feltovich, P. J., Spiro, R. J., & Coulson, R. L. (1993). Learning, teaching, and testing for Complex Conceptual Understanding. In N. Frederiksen, R. J. Mislevy & I. I. Bejar (Eds.), *Test theory for a new generation of tests* (pp. 181-217). Hillsdale, NJ: Erlbaum.

Fitts, P. M. (1964). Perceptual-motor skill learning. In A. W. Melton (Ed.), *Categories of human learning* (Vol. 47, pp. 381-391). New York: Academic Press.

French, R. M., & Cleeremans, A. (2002). *Implicit learning and consciousness: An empirical, philosophical and computational consensus in the making*. Hove, UK: Psychology Press.

Gick, M. L., & Holyoak, K. J. . (1980). Analogical problem solving. *Cognitive Psychology, 12*, 306-355.

Hayes, J. R. (1981). *The complete problem solver*. Philadelphia: The Franklin Institute Press.

Heathcote, A., Brown, S., & Mewhort, D. J. K. (2000). Repealing the power law: The case for an exponential law of practice. *Psychonomic Bulletin & Review, 7*, 185-207.

Hoffman, R., Shadbolt, N., Burton, A., & Klein, G. (1995). Eliciting knowledge from experts: A methodological analysis. *Organizational Behaviour and Decision Processes, 62*, 129-158.

Kieras, D. E. (1998). A guide to GOMS model usability evaluation using NGOMSL. In M. Helander, T. K. Landauer & P. Prabhu (Eds.), *Handbook of human-computer interaction* (Vol. 12, pp. 391-438). Amsterdam: Elsevier.

Kieras, D. E., & Bovair, S. (1986). The acquisition of procedures from text: A production system model. *Journal of Memory and Language, 25*, 507-524.

Kimble, G. A. (1961). *Hilgard and Marquis' Conditioning and Learning* (2nd Edition ed.). New York Appleton-Century-Crofts.

Larkin, J. H. (1981). Enriching formal knowledge: A model for learning to solve textbook physics problems. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition* (pp. 311-334). Hillsdale, NJ: Erlbaum.

Larkin, J. H., McDermott, J., Simon, D. P., & Simon, H. A. (1980a). Expert and novice performance in solving physics problems. *Science, 208*, 1335-1342.

Larkin, J. H., McDermott, J., Simon, D. P., & Simon, H. A. (1980b). Models of competence in solving physics problems. *Cognitive Science, 4*, 317-345.

Lesgold, A. M. (2001). The nature and methods of learning by doing. *American Psychologist*(November), 964-973.

Lieberman, M. B. (1984). The learning curve and pricing in the chemical processing industries. *The RAND Journal of Economics, 15*(2), 213-228.

Lindsay, P. H., & Norman, D. A. (1977). *Human information processing* (Vol. 2nd). San Diego, CA: Harcourt Brace Jovanovich.

Moray, N. (1999). Mental models in theory and practice. In D. Gopher & A. Koriat (Eds.), *Attention and performance XVII: Cognitive regulation of performance: Interaction of theory and application* (pp. 223-258). Cambridge, MA: MIT Press.

Neisser, U. (1976). *Cognition and reality*. San Francisco: W. H. Freeman.

Nerb, J., Krems, J., & Ritter, F. E. (1993). Rule learning and the power law: A computational model and empirical results. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, 765-770. Erlbaum: Hillsdale, NJ.

Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

Newell, A., & Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition* (pp. 1-51). Hillsdale, NJ: Erlbaum.

Ohlsson, S. (1992). The learning curve for writing books: Evidence from Professor Asimov. *Psychological Science, 3*(6), 380-382.

Pavlik, P. I. (2007). Timing is in order: Modeling order effects in the learning of information. In F. E. Ritter, J. Nerb, E. Lehtinen & T. O'Shea (Eds.), *In order to learn: How the sequences of topics affect learning* (pp. 137-150). New York, NY: Oxford University Press.

Rabbit, P., & Banerji, N. (1989). How does very prolonged practice improve decision speed? *Journal of Experimental Psychology: General, 118*, 338-345.

Rasmussen, J. (1983). Skills, rules, knowledge: Signals, signs and symbols and other distinctions in human performance models. *IEEE Transactions: Systems, Man, & Cybernetics, SMC-13*, 257-267.

Ritter, F. E., & Schooler, L. J. (2001). The learning curve. In *International encyclopedia of the social and behavioral sciences* (Vol. 13, pp. 8602-8605). Amsterdam: Pergamon.

Schmidt, R. A., & Bjork, R. A. (1992). New conceptualizations of practice: Common principles in three paradigms suggest new concepts for training. *Psychological Science, 3*(4), 207-217.

Schneider, W. (1985). Training high-performance skills: Fallacies and guidelines. *Human Factors, 27*(3), 285–300.

Seibel, R. (1963). Discrimination reaction time for a 1,023-alternative task. *Journal of Experimental Psychology, 66*(3), 215-226.

Simonton, D. K. (1996). Creative expertise: A life-span developmental perspective. In K. A. Ericsson (Ed.), *The Road to excellence: the acquisition of expert performance in the arts and sciences* (pp. 227-253). Mahwah, NJ: Erlbaum.

Singley, M. K., & Anderson, J. R. (1989). *The transfer of cognitive skill*. Cambridge, MA: Harvard University Press.

Spiro, R. J., Feltovich, P. J., Coulson, R. L., & Anderson, D. R. (1989). Multiple analogies for complex concepts: Antidotes for analogy-induced misconception in advanced knowledge acquisition. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning* (pp. 498-531). Cambridge, UK: Cambridge University Press.

Wiener, E., Kranki, B., & Helmreich, R. L. (Eds.). (1993). *Cockpit Resource Management*. London, UK: Academic Press.

Wright, T. P. (1936). Factors affecting the cost of airplanes. *Journal of the Aeronautical Science, 3*(2), 122-128.