

Components of Elements: Procedural and Declarative Components of Transfer

Marita Franzke
Department of Psychology
University of Colorado at Boulder
Tech Report #93-02

This research was funded by NSF Grant #IR 9116640



INSTITUTE OF COGNITIVE SCIENCE
UNIVERSITY OF COLORADO AT BOULDER
BOULDER, COLORADO 80309-0344



Chapter 1. Introduction	1
1.1 Common elements vs. generalized transfer.....	1
1.2 Decomposition of common elements.....	2
1.3 Overview.....	3
Chapter 2. Current Theories of Learning and Performance.....	4
2.1 Act*	5
2.1.1 Basic structure.....	5
2.1.2 Learning.....	8
2.1.3 Performance.....	11
2.1.4 Transfer Predictions	11
3.2 Cognitive Complexity Theory.....	14
2.2.1 Basic structure.....	14
2.2.2 Learning.....	15
2.2.3 Performance.....	15
2.2.4 Transfer predictions	16
2.3 Soar.	17
2.3.1 Basic Structure.....	18
2.3.2 Performance.....	18
2.3.3 Learning.....	19
2.3.4 Transfer predictions	20
2.4 Summary.....	21
Chapter 3 . Transfer Research in the Procedural Learning Paradigm	23
3.1 Support for the common elements view of transfer.....	23
3.1.1 The CCT research group.....	23
3.1.2 The Act* Research group	27
3.2 The phenomenon of surplus Transfer.	32
3.2.1 Surplus Transfer in Text Editing.....	32
3.2.2 Use specificity?.....	34
3.2.3 Summary.....	39
3.3 The case of declarative transfer.....	40
3.3.1 Declarative Transfer in operator selection.....	40
3.3.2 Dedarative transfer in subskills of Lisp programming.....	41
3.3.3 Summary.....	43
3.4 Conclusions.....	44

Chapter 4. Transfer in the Verbal Learning Paradigm.....	47
4.1 General introduction	47
4.2 Components of Learning	49
4.2.1 Learning to Learn.....	50
4.3 Specific Transfer components.....	52
4.3.1 Response Learning	52
4.3.2 Stimulus differentiation	53
4.3.3 Response differentiation.....	53
4.3.4 Forward Associations.....	53
4.3.5 Backward Associations	53
4.3.6 Summary.....	54
Chapter 5. General Conclusions	55
5.1 Procedural vs. declarative learning	55
5.2 Outlook	55
References.....	57

Chapter 1. Introduction

The goal of the current paper is to summarize and discuss the status of a common elements view of procedural transfer. Skill transfer or transfer of any kind of knowledge is seen as the phenomenon by which the previous acquisition of a piece of knowledge facilitates or hinders learning of another piece of knowledge. A simple example would be the facilitation of learning one programming language after having learned similar ones. Similarly one could imagine a negative effect of learning one set of commands to use for a computer application on the acquisition of second different but similar set of commands.

1.1 Common elements vs. generalized transfer

Early in this century (Thorndike & Woodworth, 1901) a view of transfer of knowledge was put forward that emphasized the importance of common identical elements of knowledge for any kind of transfer. This position was opposed to an older view of transfer that emphasized the role of formal discipline (e.g. Angell, 1908). The proponents of this faculty theory of transfer had a metaphorical view of the mind as a group of muscles that could be strengthened with certain formal exercises. Transfer was explained not in terms as transfer of particular knowledge of facts or procedures, but simply as a result of the strengthened power of a certain muscle group (or faculty) to perform its particular function. If a certain faculty (such as reasoning) was trained well, these theorists expected broad general transfer to any tasks that required the function of this faculty.

Thorndike and Woodworth, who came from a behaviorist background, began putting the transfer phenomenon to experimental test. From their observations they concluded that transfer could only happen as transfer of one stimulus-response association to the execution of the same association at a later point in time. In its strongest form this common elements view even excluded similar S-R associations from transfer. Even though this extreme position can be attacked on logical grounds, because no psychological stimulus situation can be exactly the same as another (and therefore transfer should never happen), the general idea of describing and investigating transfer in terms of identical elements has been very influential for subsequent work on transfer. In this process, the strict definition was lifted to include similar but not identical S-R associations (e.g. Osgood, 1949), and in more recent work, the nature of the common elements has been changed from the behaviorist construct of S-R associations to the cognitivist construct of production rules (e.g. Anderson, 1982). However, the important inheritance included the conviction that transfer could be explained solely on the background of the relevant shared elements, and not on any sort of generalized component.

In this paper we will review experimental results from two schools within this general common elements approach. We will first review relevant theories and experimental research in the information processing paradigm. In this approach each task requires the learning of cognitive procedures to manipulate symbols and structure the task. The general unit of analysis in school of thinking are rules that embody these cognitive procedures or operations. In a later chapter the transfer work of an earlier group of researchers, the 'verbal learners' will be reviewed. This group of researchers was concerned with negative transfer in form of interference between two successively learned sets of stimulus-response associations. In particular interference was studied using verbal material, such as lists of words or nonsense syllables, therefore the term 'verbal learners'. The unit of analysis here is therefore the particular S-R associations made up by certain characteristics of the word lists to be learned.

1.2 Decomposition of common elements

In both cases we will see that the experimental results show levels of transfer that could not be explained by the proposed common elements alone. In search of a neutral description of this effect, we will call this phenomenon 'surplus transfer'. Instead of arguing for the existence of generalized transfer, we will make the argument that in both approaches the analysis of what transfers started at too a general level. In both approaches an element has been defined as a learned association between two (otherwise independent) events. In the information processing approach these two events are the descriptions of a certain cognitive state that becomes associated with a specific cognitive operation. In the case of the verbal learners these events are simply the list of learned (verbal) stimuli and (verbal) responses. In both cases, transfer has been conceptualized as the whole-cloth transfer of these associations to a new task or situation. However, recent results in the analysis of procedural transfer (e.g. Singley & Anderson, 1989; and Pennington & Nicholich, in press), as well as the research of the verbal learners (e.g. Martin, 1965) demonstrate that learning of such associations may be a complex phenomenon. It can be decomposed into several stages and into the learning of several components. For example, the events that make up the new association may be new to the subject and have to be acquired as concepts before they can be associated to each other. If this is the case, partial learning (learning of the events rather the association) may lead to transfer. From a high level analysis of the learning task such transfer may look unexplainable and suspiciously general. A lower level of analysis however, can discover common elements in these components of the associated pairs of events, that may have the power to explain the surplus transfer that has been observed.

1.3 Overview

In the second chapter of this paper the theoretical background for the modern common elements view of transfer will be reviewed. As will be shown in detail, the common denominator between three general theories of learning and performance (Act*, CCT and Soar) is the conviction that procedural skill is represented in a rule-based manner. On this background, transfer is predicted on grounds of these common rules whole-cloth from one task to another. The third chapter reviews the experimental literature that has been accumulated to investigate transfer as an effect of these common elements or rules. As the review will illustrate, common 'rule' theories under predict transfer in certain experimental situations, so that the analysis of those rules into its smaller components seems indicated. Chapter four summarizes the research of the verbal learners. It will be pointed out that research in this paradigm took a similar path. The components identified by this research group will be discussed in some detail. In the general conclusions we reflect on the possible similarity between the components identified in both approaches and considers the relevance of the verbal learning research for the current analysis of procedural transfer. We conclude that in both cases, the paired elements that have to be associated (production rules and stimulus -response connections, may have to be acquired as independent events. This acquisition period is especially important in situation where these events (verbal material in one case, and representations of new cognitive operators, or new symbols that indicate a certain problem state) are unfamiliar, initially meaningless and difficult to differentiate. We conclude that initial components of learning (and possible candidates for transfer) are therefore the acquisition of semantic meaning, and differentiation between a set of new concepts.

Chapter 2. Current Theories of Learning and Performance

Before reviewing the current experimental research on transfer we first have to understand the theories that have driven this type of research. In particular we have to familiarize ourselves with the particular level of analysis of cognitive behavior that these theories have chosen. The level of analysis defines the grain size of the common elements that have been postulated to underlie transfer. Our concern here will therefore be to determine which type of cognitive elements are proposed to transfer, and how the theories explain learning and performance based on such elements. Finally we will describe the particular transfer hypothesis of each such theory.

In the information processing approach mental activity is understood as a series of actions or operations performed on some currently available mentally represented state of affairs. What type of operation is performed on what particular representation is stored in rules, that specify a certain cognitive operation for a certain representation. It is assumed that an individual has a virtually unlimited capacity to store more and more of such rules. Transfer is therefore understood on the background of the learning and transfer of such rules. These exactly are the common elements that current theories of transfer base their prediction upon. In the following sections we will review in some detail how these rules have been described, how they are acquired by a learning individual, and what transfer predictions have been derived on their background, for three different theories of learning and performance. Our main conclusion is going to be that particular transfer predictions are based on transfer of common processing rules for each of the reviewed theories (Act*, CCT and Soar). There are some differences as to how exactly these rules are represented, and how they are acquired (i.e. Act* and Soar). The main differences between these two models seems to lie in the learning paradigms that have been explored, which leads to differently general rules that are acquired. Act*, which explicitly models plausible and observed human learning events, assumes that learning happens in a fairly controlled environment, where subjects are kept on a solution path that is relatively narrow. If errors occur frequently, subjects are recovered and lead back to the optimal solution path after a few trials. Soar is an attempt at exploring problem solving and learning from a Artificial Intelligence perspective. Here the emphasis is not necessarily at modeling a plausible learning paradigm, as much as the exploration of a theoretically possible series of events that may lead to learning of a set of rules representing a particular problem type. Consequently, time and motivational constraints are not given high priority and learning may happen in an unconstrained space, that assumes exploration with minimal feedback from an instructional environment. Thirdly, we will see that the two theories (Act* and Soar) that explicitly model learning, can be extended to explain transfer on grounds of common declarative (not rule-based) knowledge. CCT, which assumes learning events as described by Act*, can therefore also principally be extended to explain transfer on this basis.

2.1 Act*

Act* is the one member in a family of ACT models that embody John Anderson's theory of cognition. It can be seen as the earliest and the most prototypical attempt at a unified theory of cognition, and has been explicitly applied to both, interference phenomena as discussed in the verbal learning domain (e.g. Anderson, 1976 and Anderson 1983), as well as procedural transfer (e.g. Singley & Anderson, 1989). The theory has been modified over successive generations, to explain new data in a more efficient way (see Anderson, 1992). This review will be based on Act* as described in Anderson 1983. With respect to transfer three important theoretical works have been published (Anderson, 1982, 1987, and Singley & Anderson, 1989). Whereas the first two papers describe the process of procedural learning and transfer in detail. In this earlier work the identified procedural elements were proposed as the one single mediator of skill transfer. The Singley and Anderson volume starts considering the case of transfer of declarative elements. We should keep in mind, however, that this is an extension to a theory that started proposing common procedural elements as the sole explanation for any transfer effect.

2.1.1 Basic structure In the ACT framework, memory is seen as containing separate structures for declarative and procedural knowledge, as well a Working Memory that consists of currently active declarative representations.

Declarative knowledge, that is knowledge about facts, is represented in an associative network, consisting of cognitive units and links between them. These cognitive units may consist of temporal strings, spatial images, or abstract propositions. An example would be a list of propositions such as

- (1) exist (office D446A)
- (2) location(office D446A, fourth floor Muenzinger)
- (3) is-a(office, place to work)
- (4) has(office, books)
- (5) has(rec.-center, door)
- (6) has(rec.-center, showers)

etc.

Each one of these propositions could be represented as a node, links between these nodes would represent the fact that these units have been processed or stored together. For example there could be links between the first two nodes, as well as links between nodes 3-4, and between 5 and 6. However, if propositions 5 and 6 if were never processed together with propositions 1-4, there would be no links between those two groups of propositions.

Working Memory (WM) is simply defined as the set of memory elements that are active at any point in time. Such memory elements might have entered WM by perception, or may be put into WM as the product of a cognitive operation. Memory elements can also enter WM by retrieval from declarative memory.

Each unit in declarative memory, as well as the links between units have certain strengths. These strength values are an expression of the frequency with which a unit or a pair of units has been processed, and determine its retrieval probability. Retrieval is achieved by a process of spreading activation through the net of nodes, along the links that connect the nodes to each other. Activity is spread from knowledge elements that are active in WM. These active elements are called retrieval cues. Retrieval cues act as a source of activation and may lead to the retrieval of associated nodes. For example, on perceiving office D446A, activation may spread to nodes linked to proposition (1) in the example above and lead to the activation of propositions (2-4). The strength of a link determines how much activation goes down a certain path. Retrieval however depends on a unit's *relative* strength, which is defined by the strength of the link between the retrieval cue and the unit, proportional to the link strengths between other units and the retrieval cue. The strengths of units in declarative LTM may decay with time, and therefore depend on the recency of the last encoding and the frequency with which an item has been processed. Activation levels are temporary and subject to rapid decay over time.

Procedural knowledge is represented as a set of production rules. Each cognitive operation is embodied in a condition-action pair. The condition side of the rule (the left-hand side, or the if-statement) contains a specific cognitive condition, under which the action side (the right-hand side, or the then statement) should apply. A simple example would be a system with a rule such as

```
IF
    the goal is to go to office D446A
    and the current position is the second Floor of Muenzinger
THEN
    go to the next staircase
    and walk up two levels
    and modify the current position is the forth floor of Muenzinger
    and set as a subgoal to determine the identity of the staircase.
```

The condition side may contain any type of description of a currently active cognitive state, such as the goal to go to a certain office, and the knowledge about a current position in the world. The action side then contains the actions that will lead to satisfaction of the goal under the specified conditions.

Retrieval of a production is achieved by a mechanism of matching working memory elements to productions. If working memory contains active memory elements (such as the goal and the current position above), the production rule above would be matched. However there might be other similar productions that are also partially or fully matched by that state of WM. Act* embodies a conflict resolution mechanism that considers certain heuristics to select one single production to be fired. As a result of firing a production the prescribed actions are executed and old WM elements may be modified or deleted, and new elements may be added to WM.

Important heuristics that are considered in conflict resolution are the strength of the productions being matched (the frequency with which it has been applied successfully before), their specificity, where more specific productions are preferred, and goal dominance, where only productions that test for an active goal can be selected. As an example for different levels of production specificity consider the following alternative production:

```
IF
    the goal is to go to office D446A
    and the current position is ?some Floor of Muenzinger
THEN
    retrieve the ?# Floor of office D446A
    and subtract the ?# from ?some Floor and put the result in the
    ?difference
    go to the next staircase
    and walk up ?difference levels
    and modify the current position is the forth floor of Muenzinger
    and set as a subgoal to determine the identity of the staircase.
```

Even though the two productions would achieve the same result, the second production is more general and could be matched if the system was on any level in Muenzinger. However, this generality come at the cost of extra cognitive operations (memory retrieval, the subtraction and the temporary storage of the difference in WM). If one assumes cognitive efficiency for the proposed system then it should choose the prior more specific production over the general one, because the same effect is achieved in a much simpler way.

Goal dominance is the key to controlling action. For example we could imagine a production of this type:

```
IF
    the goal is to go to office D226F
    and the current position is the second Floor of Muenzinger
    and the current position is in front of office D226E
THEN
    take two steps to the right
    and knock on the door.
```

Even though this production is more specific than the first of our examples, it should not fire, because it does not serve the currently active goal (which is to go to office D446A). In other words, goal dominance helps a system not to get sidetracked by production rules that offer a good match to the currently active representation of the state of the world, but would lead it away from fulfilling the currently desired goal state.

Together, these selection principles determine that optimal (fast and reliable) retrieval of productions is achieved, when specific, goal related productions have been compiled. This leads us to the learning mechanisms proposed in Act*.

2.1.2 Learning. As there are different retrieval mechanisms for declarative and procedural knowledge, there are also corresponding storage functions for each type of knowledge. In the associative network, learning is accomplished by placing a new item into working memory. With a certain probability this item will form a long-term-memory trace. Each time the same item is placed into working memory again, the strength of the cognitive units, or trace of the original encoding is increased by one.

The mechanism for procedural learning starts with general problem solving behavior. It is assumed that any system has a set of general rules that can lead to stepwise movement towards a goal in a new domain. Consider a system that has not been in Muenzinger before, but has general rules to find its way around in buildings. For example there may be general rules of the following kind:

```
IF
    the goal is to go to room ?#
    and the ?level of room ?# is unknown
THEN
    set as a subgoal to determine the ?level of room ?#

IF
    the goal is to go to room ?#
    and the ?level of room?# is known
    and the current position is unknown
THEN
    set as a subgoal to determine the current position

IF
    the goal is to go to room ?#
    and the ?level of room ?# is unknown
    and the subgoal is to determine the ?level of room ?#
THEN
    read the next character of ?#
    and put it into ?char
```



```

IF
    the goal is to go to room ?#
    and the level of room ?# is unknown
    and the subgoal is to determine the level of room ?#
    and the ?char is a numerical
    and the ?char is not a letter
THEN
    make the level of room ?# is known
    and put the ?char into ?level
    and delete the subgoal to determine the level of room ?#

etc.

```

Act* builds its learning mechanism on the execution of such general problem solving rules. It assumes that a system have access to such general rules (or even more general ones, as means end analysis or the use of analogy; for a detailed example see Anderson, 1986), and slowly accomplished the general goal under frequent use of declarative knowledge (for example the representation of the office number) or cues given by the environment (for example signs that contain level- or wing- names). This phase of learning a new skill is called the *declarative* learning phase because of its dependency to access to a declarative representation of the problem.

Learning in this phase is accomplished by the *compilation* of new domain specific rules from successful execution of general rules. For example, we could imagine a new domain specific rule for parsing the office numbers of offices in Muenzinger.

```

IF
    the goal is to go to room ?#
    and room ?# is in Muenzinger
    and the ?level of room ?# is unknown
    and the subgoal is to determine the ?level of room ?#
THEN
    make the ?level of room ?# is known
    and put the second char of ?# into ?level
    and delete the subgoal to determine the ?level of room ?#

```

This would save the system one additional processing cycle and make the search somewhat more efficient. Act* assumes that new, more specific rules like this are compiled at all parts of the problem solving process, during the first trial of successful learning. The product of this learning phase then is a set of domain specific rules, that let the system find the room much faster in its second trial.

In Act* this ends the phase of declarative learning. Additional learning trial on this same rule set lead to proceduralization and composition of the new rules.

Proceduralization is the learning event in which additional new rules are compiled from the original set of rules that are even more specific than the original set. For example we could imagine that a specific rule for retrieving the level of office D446A is formed:

```
IF
    the goal is to go to room D446A
    and room D446A is in Muenzinger
    and the ?level of room D446A is unknown
    and the subgoal is to determine the ?level of room D446A
THEN
    make the ?level of room D446A is known
    and put 4 into ?level
    and delete the subgoal to determine the ?level of room D446A
```

Composition then composes two or more rules serving the same general goal into one single rule that accomplishes the same result. For example two rules for determining the level and the wing of room D446A may be combined into one rule determining its position:

```
IF
    the goal is to go to room D446A
    and room D446A is in Muenzinger
    and the ?level of room D446A is unknown
    and the ?wing of room D446A is unknown
    and the subgoal is to determine the position of room D446A
THEN
    and put 4 into ?level
    and put D into ?wing
    make the ?level of room D446A is known
    make the ?wing of room D446A is known
    and delete the subgoal to determine the ?level of room D446A
```

Recently Anderson (1987) claimed that the process of *compilation* will always produce a new general production, a production that is variabilized and in which the specific elements to be used still have to be retrieved from declarative memory. However, it is likely that the particular production learned (general or specific) depends on the type of learning situation involved. If learners are encouraged (as in Anderson's experimental situations where learners interact with a tutor, and worked examples), to learn by example or by analogy, or by exploration, it is likely that people indeed accrue general productions from the comparisons between various instances of an application. If learners however are simply told, what is appropriate in a certain situation, they might acquire very specific productions. (In our example a learner that is taken by the hand and led to room D446A from a specific starting position, might learn a set of simple and specific productions that enable him to find his way back to the same room. In this way he might not have learned anything about parsing the room

number as a help for orientation). Whatever the specificity of the production learned, there is a sharp and marked improvement in speed and accuracy from solving one to another problem of the same kind. Where the problem could only be solved interpretatively, that is under heavy use of declarative knowledge and general problem solving procedures, the second trial profits from the use of an already compiled domain specific production. The two additional processes of further learning processes substitute variable names with constants proceduralization (substitution of variable names with constants) and composition (the combination of two sequential atomic productions into one larger macro-production) lead to further fine-tuning of the additional rule set. It is assumed however, that more specific rules do not replace the more general rules, but simply add to them. In other words, the more general rules acquired during initial problem solving are still available if a different problem in the same domain arises (for example of the general parsing rules for indications of locations of other offices in the same building). In other words, once a procedure exists, more specialized versions of it can be learned easily and compact rules can be established that lead to the rapid execution of many successive actions. Furthermore, the strength of an existing production will be increased with each further successful application. This also adds to the increased efficiency of an well learned rule set by decreasing the time needed to fire the appropriate rules.

2.1.3 Performance. Skilled performance is conceptualized as the straightforward application of a such a domain specific rule-set. Given that a top-level goal statement has entered working memory and a set of propositions describe the state of the world, the procedure that most closely matches this combined set of productions, and that is most specific will be fired. If the task has a complicated structure (for example text editing), there will be control productions, that fire subgoals. These subgoals will enter working memory and (together with the propositions encoding the state of the world) trigger productions that work in fulfilling the particular subgoals. This process will continue, until all subgoals and goals have been satisfied, and the task is fulfilled. Speed of performance is therefore a function of the number of productions that have to be matched as well the time required for a particular selection. A specific and strong set of productions will lead to fast and unambiguous selection in each cycle.

Anderson 1983 and 1987, as well as McKendree and Anderson (1987) discuss empirical evidence for these type of learning mechanisms. Since this paper is concerned with transfer we will limit the discussion of empirical evidence to studies directly concerned with transfer.

2.1.4 Transfer Predictions. *Procedural transfer.* Procedural transfer in the original proposal (Anderson, 1987) is essentially a modern version of a common-elements theory of transfer (Thorndike & Woodworth, 1901). Transfer to a new task is expected to the degree by which the new task can be achieved under the use of already acquired domain specific productions. Transfer is therefore dependent on the overlap between two needed rule sets,

and on the generality of the rules that were acquired during first task learning. In terms of our example, the student should show large transfer from the first task to finding room D346A, given the same starting position, but lower transfer to finding the same room when given a different starting position that is not on the correct solution path. For example, if procedures for directing locomotion in the building are represented in terms of visual cues that can only be seen when starting in a certain position, and the student is started in a different corner of the building, then the same specific rules simply do not apply and can not help in finding a given office. In that case only general rules for parsing the room number for directions may apply and lead to moderate amounts of transfer from one task to the next. Making predictions about transfer is therefore critically dependent on a accurate task analysis that captures exactly how subjects encode and represent the newly learned skill and the transfer task.

Negative transfer would be observed either when a series of productions that was optimal for one task context will be used in the transfer task and therefore prevent a better production chain from being learned (transfer of non-optimal methods). Another form of negative transfer would be the firing of a totally inappropriate production, because of a match to its conditions (interference). This second type of negative transfer leads to errors and subsequent corrections, and should therefore be of transitory nature and only be observed in early transfer performance.

Another interesting prediction that can be derived from this definition of procedural transfer is what Singley and Anderson (1989) call 'use specificity'. This specifies the situation by which different uses of the same declarative knowledge do not transfer to each other. An example for this counterintuitive prediction would be that declarative learning the layout of Muenzinger in the context of one searching task does not transfer to finding a different room, if there is not overlap in the productions that have been compiled during its acquisition. Other examples that have been investigated are the prediction that selecting logic or calculus operators does not transfer to their application (i.e. Singley & Anderson, 1989, or Pennington & Nicholich, in press. It is exactly these types of tasks that have produced transfer results that are not compatible with a transfer theory that is based on common production transfer itself. We will come back to this point when discussing the relevant literature.

Declarative transfer. "Haunted by more transfer than (they) could readily predict" (Singley & Anderson, 1989, p. 197), modify the early assumptions that transfer could only be expected on grounds of shared productions. Here they make the concession that "there should be an initial period of positive transfer between tasks to the degree that the tasks share a common declarative base" (Singley & Anderson, ipso). This is conceptualized such that during the knowledge compilation phase, when general problem

solving procedures are applied using a declarative representation of the problem domain, access to a well practiced declarative representation should lead to some transfer between tasks. Singley and Anderson are not specific about what kind of improvement one would expect out of this type of transfer (reductions in terms of speed of access, or reductions in errors), or what type of declarative learning should lead to better transfer results.

There is also an argument to be made that declarative transfer can be expected also after the compilation phase of learning, or when domain specific but non proceduralized productions do the work.

Let's consider, how ACT* would explain negative transfer in terms of the interference effects observed in the verbal learning literature (for details see below). Anderson (1983) describes the effect of interference as phenomenon of retrieval from declarative memory, as what he calls the 'fan effect': Given a retrieval cue, it will take longer to retrieve an item from declarative memory, when the memory unit representing the retrieval cue has links to many other units that have been stored together with it, as if the retrieval cue has only been linked up to one or few other concepts. In Anderson's 1983 theory, the retrieval is controlled by productions that probe memory and recognize an item as retrieved, but the retrieval itself is handled by spreading activation in declarative memory. If many links emanate from a node representing a retrieval cue, proportionally less activation is spread down every single link, and the retrieval of the relevant item will take longer. Declarative interference during the execution of a procedural task would be explained in a similar way. Recall, that in most cases people's representation of a skill will be compiled, but not necessarily be in proceduralized form. This means that the productions encoding a new skill will have variables for particular actions. Retrieval of the values for these variables (a particular keystroke, for example) may depend on declarative memory and is therefore subject to possible interference effects, if the retrieval cue used (the state of other conditional statements, or the state of the world) do not serve to identify the value unambiguously. Consider the following example.

```
IF
    the goal is to call ?somebody
THEN
    retrieve ?somebody ?telephone-number
    and set the subgoal to dial ?telephone-number.
```

Given, that ?somebody has already been instantiated as Otto, Otto would serve as retrieval cue for his phone-number. Unfortunately, Otto has moved 5 times in the last two years, so there are 5 links pointing from Otto to 5 different phone numbers, in declarative memory. The declarative retrieval process might therefore come back with the wrong information, or may

simply take a while, but the production would simply take that information and execute it blindly . This example demonstrates that negative transfer in the form of an action slip could have its roots in declarative interference. It also demonstrates that additional declarative practice in memorizing Ottos new phone number and distinguishing it from his old numbers might lead to positive transfer, that is an addition to transfer that can be expected on grounds of using the compiled rules for retrieving phone numbers and dialing.

To summarize this discussion. Originally (Anderson, 1983, 1987) predictions about transfer based on Act* were exclusively derived on the background of a shared set of productions. The particular specificity of these common elements determines how narrow or broad the predicted transfer will be. Recently (Singley & Anderson, 1989), a well practiced base of declarative knowledge is seen as a possible source of transfer during the initial stages of learning. We illustrated the second prediction by showing that declarative knowledge can act as a source of positive and negative transfer. Declarative transfer of this type can be expected as long as the productions used in the transfer task are general enough to make some sort of access to declarative knowledge. We would not expect to see declarative transfer in a task that profits from use of fully compiled, proceduralized and composed production rules.

3.2 Cognitive Complexity Theory

We will now turn our attention to a model of procedural performance and transfer, that is very similar in spirit to the original Act* account of transfer. Cognitive Complexity Theory has been developed independently by Kieras and Polson (1985), mainly to explain learning and transfer effects in the area of human computer interaction. The theory was never intended as a general theory of cognition, and therefore has a much smaller scope than John Anderson's Act*. However, transfer research based on this theory has been quite successful (see review in the next chapter).

2.2.1 Basic structure. Cognitive Complexity Theory (CCT) is a purely procedural theory that attempts to describe the interactions between a (simulated) user with a (simulated) computer system. The theory does not make any assumptions about the existence or form of a separate declarative memory store. Therefore it's two components are a production system to embody the user's cognitive states during the interaction with a system, and a Generalized Transition Network that implements the system behavior. This GTN performs certain system actions given (simulated) user input and its current state, and produces feedback to the user. Since our concern here is with human learning and transfer, we will focus our review on the particular structure of the production system architecture used. The general structure of

CCT's production system component is similar to the production system notation used for Act*. However, there are some noteworthy differences:

First, CCT does not have a conflict resolution mechanism. Instead, CCT relies on the specificity of its productions alone to select the appropriate action for a given combination of working memory elements (Bovair, Kieras, & Polson, 1990). Second, CCT is an explicit attempt at modeling a theory of human-computer-interaction (GOMS, Card, Moran and Newell, 1983), that places a high importance on the cognitive maintenance of hierarchical goal structures to control action. Therefore, CCT makes condition elements specifying goals and subgoals the main elements controlling the execution of action. Note that this is a decision relying on task analysis and assumptions about cognitive encoding and representation, rather an architectural decision. The condition elements could be rewritten in a way to allow for greater influence of encoded world-states. In the particular GRAPES implementation of Act* that was used to model transfer in Singley & Anderson, 1989, the same decision concerning the GOMS model was made. Third, CCT does not rely on partial matching. Every conditional element of a production has to be met for a rule to fire. Forth, CCT does not include different strength values for each production. Fifth, retrieval from declarative LTM is not modeled.

2.2.2 Learning. CCT is based on a range of learning assumptions: Initial learning of procedures is a product of general problem solving processes, in the manner of Act* (Kieras & Polson, 1985). Learning of new rules requires a constant amount of time, no matter which learning processes would have to be postulated for learning (Bovair, Kieras, & Polson, 1990). New rules contain constants: no variabilization of parameters in the beginning learning stages. Novices check feedback explicitly: this leads to the implementation of extra feedback checking production rules for each new acquired step.

In modeling the transition of a novice user to an expert user, CCT assumes that practice makes the novice rule set more compact and more general. (Act* makes the opposite prediction that practice makes the rules more specific). In particular, an experts procedural skill is derived from a novice rule set by: (1) collapsing rules (similar to composition in Act*) (2) removing the feedback-checking rules.

2.2.3 Performance. Novice or expert performance on a task is a straightforward execution of the simulation model described above. Predictions about performance time are derived by the following function:

Execution Time = # of cycles X time per cycle + time spent in operators

The Number of cycles can be derived simply by counting the numbers of rules in the model. It is assumed that all recognize-act cycles take the same amount of time to execute, but this is a free parameter that has to be estimated

from the particular data of a certain task. From previous research it can be expected that the approximate time for a cycle lies in the range of about one tenth of a second per cycle (Bovair, Kieras, & Polson, 1990; Card, Moran & Newell, 1983). Time spent in operators is another constant that has to be estimated from the data, and is related to the physical aspects of performing certain actions. The difference in performance time between novices and experts would therefore simply be explained in the fewer number of recognize-act cycles that need to be executed (no feedback, larger rules). Using multiple stepwise regressions to model individual performance times for editing tasks, most of these predictors have explained a significant amount of variance in the data (Bovair, Kieras, & Polson, 1990): To summarize the results, the analysis provides clear evidence for the effect of number of verification steps on individual performance time. Furthermore there is a significant interaction with level of expertise such that low levels of expertise are associated with more time needed to verify the outcome of actions. Also, the number of recognition-action cycles derived from a task decomposition particular to the level of expertise explained are large part of the variance. Unfortunately this predictor variable (number of cycles) was highly correlated with the number of keystrokes, so that the relative effect of each variable could not be determined.

2.2.4 Transfer predictions. The transfer predictions of CCT are straightforward, and overlap partially with the procedural transfer predictions discussed in the context of Act*. It is assumed, that old productions transfer to a new task to the degree that they overlap with the new task. Learning time on a novel task is seen as the function of the number of old productions and some constant associated with the performance of all other involved cognitive and motor operations. Note that this constant can theoretically be further analyzed into the predictors that were found to be relevant in the analysis of (non-learning) performance times, as discussed above (Bovair, Kieras, & Polson, 1990).

A set of additional assumptions has to be made, however, because CCT's learning assumptions produce quite specific (unvariabilized) productions. First, if a new task uses exactly the same rule as a previously learned one (i.e. 'delete string' - 'delete string') the appropriate rule will fire and no additional learning time is needed for execution. Second, if a new task affords the use of a rule *on the same object*, but necessitating a different action (i.e. 'delete string' - 'modify string'), the old rule will be *generalized* by variabilizing all verb terms (in the conditional elements and the action elements) in the rule (i.e. 'delete string' --> 'X? string'; Bovair, Kieras, & Polson, 1990). In some cases, generalization of the object instead of the verb component is indicated. This would be the case, for example, when transfer is observed between deleting different types of objects. If generalization is necessary within a particular task, a small amount of extra learning time is

expected. Finally, if a task affords the use of an already generalized production, no additional learning time is expected for this production.

In comparison to Act*, which shows skill acquisition as the movement from initially general productions to the addition of more and more specific productions, CCT makes the opposite prediction: This theory assumes that productions are first acquired in a very specialized way, and become generalized with broader use.

This, however should probably not be decided in such a general way: As mentioned above, there might be learning situations (learning by being told, without use of analog examples that facilitate the generalization across classes of objects or concepts), in which subjects indeed move from very specialized representations to more generalized representation after exposure to different cases. This would be the case when minimal declarative knowledge is presented to subjects, who then follow procedural instructions blindly. The more the learning procedure facilitates problem solving or exploration, the use of example problems, toy problems, or analogies, the more general should the initial representation of the procedural skill be. The particular type of initial representation of a skill should therefore rather be guided by careful tasks analysis from case to case than by rigid architectural assumptions.

2.3 *Soar*.

Soar is a cognitive architecture developed by Laird, Rosenbloom and Newell (1986) in an attempt to build a unitary theory of cognition, based on the notion of problem spaces as the fundamental category of cognition (i.e. Newell, 1980). Because Soar's general Architecture is critically determined by this general assumption, the problem space hypothesis will be summarized first.

A problem space is a collection of states that can be traversed in order to reach a goal state. It is defined by what goal state is selected, by the current state of the problem solver, and by the operators that define the transition of current states towards the goal state. For example, the goal is to find office D 446 A in the Muenzinger Psychology building. The respective problem space could be defined as 'finding your way around the Muenzinger Psychology building', defined by all possible states in the Building, and all the operator that move a person around in these states. A current state could be 'in front of the main office'. A set of possible operators would be 'turn right', 'turn left', 'walk upstairs', 'walk downstairs', 'parse office number for directions' etc. An operator that is obviously not relevant for this problem space is 'pour the cereal into the bowl'. Applying any of the legal operators will transform the current state in any of the possible states within the problem space. For example 'turn left' will transform the state 'in front of the general office' into 'at the intersection of the E and D wing on the second floor'. It is important to

notice that this notion of the problem space is a logical notion, not an assumption about an active cognitive representation. At any point in time, a problem solver does not activate all the states and operators that are defined by the problem space it is traversing, but it will activate a subset of operators and states, depending on its current situation in the problem space. This current situation is defined by the already traversed states and operators used, her current state, and the goal state. It is assumed that knowledge of this form is active in working memory. In an elaboration phase this set of knowledge will retrieve all associated relevant information from Long Term Memory, and in a decision process, the next step will be selected. At this point in the discussion of problem spaces the particular structure of Soar becomes relevant

2.3.1 Basic Structure. Different from Act*, Soar is an attempt to model all cognitive phenomena within one general memory structure. All different types of memory units, procedural, declarative, episodic and semantic are encoded as productions in a unitary memory. This production system is thought of as a content-addressed memory. The conditions of productions specify retrieval cues, which call the right-hand side into working memory when they have been matched. This retrieval process acts in parallel, and there is no conflict resolution in the traditional sense, so all productions that are related to a set of WM elements will be activated and also called into WM. This retrieval process continues until quiescence or until no further associated productions are matched. Note that this process is not much unlike the spreading activation mechanism discussed in the chapter on Act*. Associations are not encoded via nodes and links, but via a production that establishes the link between concepts in the left-hand side and concepts on the right-hand side. There are important differences, however: (1) In this notation, productions are all or none: Either all of their conditional elements are matched and they will fire, or they will not fire. (2) Also, the retrieval links are asymmetrical, the left-hand side can be matched to retrieve the right hand side, but not vice versa. In ACT*'s associative networks the links can be traversed in both directions. (3) The productions serve to retrieve either declarative or procedural information. In Act* the associative network serves to retrieve declarative knowledge with spreading activation, whereas the production system handles retrieval of procedural knowledge. (4) The representation of units of memory in Soar is determined by the problem space metaphor discussed above. The key elements (objects) that constitute the right-and left-hand sides of productions are goals, problems spaces, states, operators, and preferences. World objects are related to this representation as specification of states and operators.

2.3.2 Performance. Since learning is a logical consequence of performance in Soar, we will first summarize how a Soar system would perform a given task. In Soar performance is problem solving. Problem solving is defined as the transition of a problem space defined by a goal. In theory, given a goal, a problem solver will select an appropriate problem space, then an appropriate initial state, and then an appropriate operator to transform the current state into one that is a shorter difference away from the

goal state, than another operator etc., until the goal state is reached. Selection of problem spaces, states and operators is in part achieved by the retrieval (*elaboration*) mechanism described above. Retrieval, however, only leads to the activation of many related productions in working memory, it does not *select* the one production, that will move the problem solver ahead in its traversal of the problem space. This selection will be achieved by an additional process, the *decision procedure* which evaluates a set of memory elements, the *preferences*, that have been added to working memory during the elaboration phase. Preferences are propositions that suggest whether a given object is an acceptable or a poor choice, given the current context, and should lead in the best case to the unanimous selection of one object to fill the current empty slot. This would be the case for rote performance of an expert, who knows exactly what to do to achieve a certain goal, given a starting state. However, there are many cases in which the decision procedure cannot select an object for use. These situations are called *impasses*. There are different reasons for such *impasses*: (1) There may be many preferences for different objects, but after all of them have been evaluated, no object stands out as the one to be preferred [tie impasse, and a more specific type of this: conflict impasse]. (2) The problem solver has only negative evaluations for the retrieved objects, or no preferences (experiences) at all: The result is that no choices are available [no-change impasse]. (3) The only preference retrieved advises to reject a prior decision. This situation is defined as *impasse*, because it has the potential to produce an endless loop [reject impasse]. Whenever an *impasse* is reached, a goal will be created to overcome the *impasse*. As a response to this, a new problem space will be selected, with new a current state and so on, so that search will now be redirected to a new problem space that is an expansion of the original one, on a lower level. In this problem space *impasses* can occur again, so that another problem space will have to be traversed, and so on. Problem solving is therefore the traversal of a hierarchy of nested problem spaces toward the original goal. This leads the discussion to the question of learning.

2.3.3 Learning. The access to preferences, and the success of a set of preferences to constrain a situation unambiguously depends on previous experiences with certain choice objects in certain contexts defined by problem solving histories. Soar embraces a continuous learning assumption. This means that during the solving of any problem at any level in the problem solving hierarchy, new productions and preferences can be accumulated, or *chunked*. . Chunking occurs, whenever an *impasse* has been resolved, that is whenever a subgoal has been satisfied. This happens, whenever a preference is produced that resolves the decision process at the next higher level. All elements that lead to a resolution of that conflict, will be backtracked along the retrieval connections that prove which elements in Working Memory before the *impasse* occurred led to the retrieval of the solution. Once these elements have been identified, they will be chunked into a new proposition. If exactly the same situation should be encountered again, this new production will fire, and resolve the conflict in the same manner. It is

obvious that this would lead to a great improvement in performance time. All learning in Soar relies on this general chunking mechanism.

2.3.4 Transfer predictions. Procedural transfer has not been a systematic research interest within the Soar community, but general discussions of transfer mechanisms within the Soar framework can be found throughout the literature (i.e. Laird, Newell, & Rosenbloom, 1986, ps. 260-261; Laird, Rosenbloom, & Newell, 1987, p. 50-51; Steier et al. 1987; Newell, 1990, p. 318-319; and Rosenbloom, Laird, Newell, & McCarl, 1991, p. 311). All these considerations generally agree on the notion that transfer within Soar is basically a common elements theory of transfer (as Act* and CCT), and that transfer can be observed within-trial, across-trial, and even across-task. Important for all these incidences of transfer is the natural implicit generalization mechanism that is inherent in Soar's chunking mechanism. Since Soar uses backtracking to determine which working memory elements were causally connected to producing a certain outcome in a subgoal, only *problem relevant*, not task specific elements will be encoded in the newly built chunk. Therefore it can be expected that the same chunk will apply in a similar situation (and replace the spawning of a subgoal), even if some of the working memory elements are different in this new context.

Even though this common elements approach to transfer is one repeatedly discussed in the literature, another member of the Soar community, Richard Young, sees Soar's contribution to the explanation of transfer effects in a somewhat different perspective (Richard Young, personal communication with Peter Polson, 1992):

"Soar would predict less [common elements] transfer than [Act and CCT]. The kind of chunks which would fire at top level, without any impasses, are likely to be very specific chunks, and therefore they probably wouldn't apply to a situation outside of their own specialty. Much more likely is that the knowledge base from which those top-level chunks were derived may be as relevant to the transfer task as it was to the original task, so the transfer will consist of rapid acquisition of a new task to a level possibly comparable with the expertise on the old task. Thus, [this is] a story based on shared knowledge but NOT on common (top-level) elements".*

To make this comparison between Act*, CCT and Soar more explicit, I would like to map 'elements' in Soar to the elements in the other two theories. In effect, what is represented at the top level of a Soar hierarchy, is the rote performance of a sequence of steps, given a top level goal and a current situation. Soar assumes that this state of affairs is reached, either when Soar is told the correct step at the spawning of a second order problem space, or when the system has exhaustively explored and learned all the correct choices to be made bottom up. In CCT or Act* this would correspond to a fully compiled, proceduralized and composed skill, a skill that can be

performed in a stable environment in exactly the same way over and over again. This would be skilled rote performance at its extreme. All theories under discussion would assume that transfer based on this type of a specialized production would always be highly specific, and would more likely lead to negative than to positive transfer. In all theories (if we assume that the learner did not acquire the skill as a rote procedure, without serious problem solving efforts), one would have to assume, that a long series of more general productions, and productions representing piecemeals of the action chain must have been acquired in the process to final mastery. It is those smaller and more general pieces of procedural knowledge, that can be expected to transfer broadly to similar tasks. How general the knowledge is that has been acquired on this path to expert performance, depends on the learning environment that was guiding the path of acquisition. If this learning environment is highly reactive, and feedback and correction is provided in close proximity to the desired overt actions (or the top level problem space for Soar), then the learner will not learn general heuristics related to the general problem type. If however the learning environment stresses the acquisition of general problem solving heuristics for certain classes of problems (which translates into the acquisition of low level problem spaces), then broad transfer should be expected. If one wanted to map the skills that have been modeled for transfer predictions in either Act* or CCT into a Soar hierarchy, they would not simply show the transition of one top-level goal space. Instead, they would show a path through the hierarchy at some intermediate level. The goodness of the match would depend on the particular learning regime used, and the degree to which learning was carried in both types of simulations.

To summarize this argument: (1) We showed that what is acquired in lower level problem spaces is not just simply 'knowledge', but knowledge cast in general, learned productions (elements). To the degree that a new problem can be solved under use of these lower level general elements, transfer can be expected. (2) Neither CCT nor Act* are restricted to modeling transfer as the transfer of a top-level specialized set of rote-performance productions. However, the tutoring environment chosen for experimental investigation does not lead subjects to explore the full hierarchy of logically possible problem spaces, and therefore the learned (and modeled) production systems (elements) are cast at a less molecular level.

2.4 Summary.

What are the important predictions about transfer that can be drawn about transfer from these theories?

(1) All reviewed models predict transfer in form of shared productions. The predictions that can be derived from the early Act* (Anderson, 1987) model and from CCT are both based on the idea that transfer to a new task is a

function of the number of known production rules and the number of new rules that have to be learned. Productions at any level are expected to transfer: Productions that control the high level goal structure of a task are expected to transfer as much as low level productions that lead to simple functions like memory retrieval or execution of motor operators. Furthermore, productions at any level of generality can transfer from one to another task. Soar's transfer predictions are based on the number of shared chunks and preferences between one and another task.

(2) Act* (Singley & Anderson, 1989) allows for the possibility of declarative transfer: Act* describes different stages of skill acquisition in detail from the beginning declarative phase, to the compilation and refinement of procedures. At any stage in this process where access to declarative knowledge is crucial, a well practiced and elaborated declarative base should benefit transfer. After proceduralization there should be no effect of declarative transfer over the transfer of specific productions. In Soar there is a chance for transfer at each point where an impasse is reached. Here transfer can be understood as better practiced productions for interpreting declarative knowledge in Working Memory. For example, it might have been learned that a certain concept held in a comprehension or situation model space is associated with a certain primitive interpretative action. This type of learned knowledge may be similar to a better elaborated representation of concepts and may lead to a faster and more effective transition towards the top-level goal.

(3) The specificity/abstractness of procedural knowledge acquired depends on the particular task, and the learning environment provided to the learner. In principal, Act* and Soar (as well as CCT theoretically) all have the capacity to learn a task in a shallow or in a exploratory, problem solving manner. The kind of learning environment provided to a student and modeled in a system should determine the breadth of transfer more than the particular architecture used to simulate it. An appropriate task analysis should therefore take the particular learning situation into account, and be guided by the learning and performance behavior that is displayed by a learner in a particular experimental situation. Soar's emphasis on transfer guided by learned productions in lower problem spaces also puts its finger on a possible neglect by the Act*/CCT community: Even if learning is driven to a point where transfer can be expected on the base of high level (specialized) common elements, general low level productions acquired in the process of learning do still exist. These productions however are frequently never modeled or included in performance models, and can hence not be used to explain additional transfer.

In the summary of the previous chapter we concluded that all current theories of learning and performance that we reviewed base their transfer predictions on some form of a common elements theory of transfer.

3.1 Support for the common elements view of transfer.

3.1.1 The CCT research group.

In a series of experiments, researchers from the CCT research group established wide support for a common elements theory of transfer based on overlapping productions as common elements (Polson, Muncher & Engelbeck, 1986; Ziegler, Hoppe, & Fähnrich, 1986; Polson, Bovair & Kieras, 1987; Polson, Muncher & Kieras, 1987; Foltz, Davies & Polson, 1988; Lee, Polson & Bailey, 1989; and Engelbeck, 1989). These studies were concerned with transfer effects in the domain of human computer interaction. Transfer was observed for skilled use of a range of different software applications, such as text editors, utility systems etc., and different interface styles (See Table 1 for a summary). As the table also illustrates, these studies have demonstrated transfer within and across task and domain, and even across applications.

3.1.1.1 Experimental Procedure. The research in the CCT group has been driven by one standard experimental procedure, which we will review briefly here. The main unit of measurement taken is task learning time to criterion. Subjects first receive some general information about the task, and the material involved in the experiment, such as the software application. Then they are trained on the sequence of tasks in a standard manner: They are presented with the task, and have to anticipate the correct actions toward the goal. If they make an error they receive immediate feedback from either the experimenter or a tutoring system. Upon a second error, they are again provided with feedback and with instructions about the correct next step. This gives the subject the chance to limited self directed learning, without the risk of losing them in an unconstrained problem space. Each task within these designs has to be performed a pre-set number of times without error, in which case it is counted as 'learned to criterion'. Measurements are taken in terms of time to reach this criterion for each task in the experiments. Assuming that each production takes the same amount of learning, this achieves a level of learning of new productions that is the same across tasks.

Recall that CCT's main transfer prediction is based on the hypothesis that learning time is a function of the number of new rules learned for a given task, and some constant related to the execution each production in each task. The goal of most of the studies from this group has been consequently, to show that training time for a given task can be predicted by

Authors	Domain	Interface style	Transfer	learning time per new rule	execution time per rule	intercept	explained Variance	Special results
Polson, Muncher, Engelbeck, 1986	Utility tasks	Menu based	Across tasks within domain	27.2 sec	n.a.	91 sec	88%	Additional Learning parameter for generalization 3.5 sec
Ziegler, Hoppe, Fährnich, 1986	Drawing, Editing,	Display based	Across tasks across domain across application	18.8 sec	n.a.	195 sec	90%	generalization across interface objects
Polson, Bovair, Kieras, 1987	Editing	Screen Editors, Command based	Within task within domain across applications	39.6 sec	n.a.	7.91 min	77%	
Polson, Muncher, Kieras, 1987	Editing	Screen Editors, Command based	Within task, within domain across applications, interference design					Transfer positive even in maximal interference condition A-B, A-Br
Foltz, Davies, Polson, 1988	Word processing	Menu based	Within task and domain across applications command synonyms, addition and deletion of steps	18.0 sec	3.6 sec	0	85%	New steps are learned as simple new rules, no lexical generalization
Lee, Polson, Bailey, 1989	Oscilloscope	Display based, Touch screen	within domain across tasks within application	9.3 sec	2.4 sec.	?	77%	
Engelbeck, 1989, experiment 1	Utility tasks	Menu based	Across tasks within domain, within application	26.9 sec	n.a.	84.1 sec	88%	

Table 1: Summary of transfer experiments from the CCT research group.

the number of proposed new rules. To show this, tasks are typically analyzed in form of a GOMS analysis, and this analysis is formalized in production system format. With the help of these models, the number of new rules for each task can be calculated easily. Experimental results in form of mean learning times to criterion are then predicted in simple linear regressions, with 'number of new rules' as predictor variable. In some cases individual training times are predicted, in which case a second predictor for individual differences is included. If common productions indeed transfer from one task to another, then the number of new rules (which have to be learned from scratch) should predict learning time perfectly and explain most of the variance in the data. In fact, this is exactly what the experiments show:

3.1.1.2 Results. Table 1 demonstrates the success of this model to account for learning times: There are four studies that predicted learning times using only one predictor variable, learning time per new rule (Polson et al., 1986; Ziegler et al., 1986; Polson et al., 1987; and Engelbeck, 1989). These studies explained between 77 and 90% of the Variance, and two studies that introduced an additional predictor variable, execution time per rule, explained 77 and 85% of the Variance. The learning times per rule that were estimated in these studies vary between 9.3 to 39.6 seconds, and there is weak evidence that the learning times are related to the interface styles subjects were dealing with: The two studies that used display based interfaces (Ziegler, et al. 1986, and Lee, et al. 1989), produced shorter learning times per rule (18.8 and 9.3 seconds) as the studies that used menu based or command based interfaces. There is one exception to this observation, however: The Foltz et al., study produced relatively short learning times (18 sec per rule) despite its use of a menu based word processor.

Furthermore, these experiments demonstrate that the same simple model can be applied to explain transfer in fairly different situations: In three experiments subjects used the same system to learn a set of different tasks (such as traversing between menus or loading a diskette) within the same general task domain and the same application (Polson et al., 1986; Lee et al., 1989; and Engelbeck, 1989). Another set of experiments addressed transfer of the same task to a different application with different low level command sequences (Polson et al., 1987a; Polson et al, 1987b; Foltz et al., 1988). Finally, one experiment addressed transfer of skill in using similarly structured command across task domains (copying, deleting, and changing the format of objects in a drawing and a word processor, Ziegler et al., 1986).

Three additional conclusions can be drawn from some of these experiments. (1) For the data from the Polson et al. (1986) experiment a second analysis was published in Polson (1988). This analysis that explained 70% of the data of individual subjects included parameters for individual differences, number of new rules, and number of newly generalized rules. It was revealed that a significant effect of number of newly generalized rules on learning time that was different from the learning time for new rules (3.5 vs. 28.7 seconds). This result is interesting in showing that the learning process

for this type of 'similar' new rule is less involved than the learning of a dissimilar new rule, but takes more time than simply executing an old rule.

(2) Two experiments investigated whether the maximal interference design of the verbal learners (A-B, A-Br) would have a similar effect on the execution of productions. In one study (Polson et al., 1987) no lack of transfer was found between two text editors that required the use of a set of command keys, that were bound to different functions for the second version of the editor. A second experiment, however (Engelbeck, 1989, second experiment) tested whether infrequent inconsistent methods (A-B, A-D) lead to retroactive interference effects after some delay. Subjects who learned an inconsistent method took made many errors in original learning of this method (and in 94% of the cases this was the false use of the typical method learned before). During retraining, subjects that learned variant methods, took a longer time to criterion and made more errors than subjects who learned prototypical methods. Again an analysis of the errors showed interference of the more frequently trained and applied typical method as the source of their difficulties.

(3) An experiment that was designed to show lexical generalization (Foltz et al., 1988) in effect demonstrated that transfer does not happen if a command name has been replaced by a synonym. In this experiment, subjects first learned to select a menu choice named 'delete' to erase a document. The transfer analysis shows that the learning time for this task in a new system where the menu choice was termed 'discard' could only be predicted if this was treated as learning of a new rule.

3.1.1.3 Summary. Taken as a whole, these results suggest that transfer in the investigated situations is well predicted by straightforward common production models. In all described experiments, training time could be sufficiently explained by the number of new rules that had to be acquired. However, transfer of low level operators seems to be fairly sensitive to similarity of source and transfer task environment. If the new task requires the use of an old rule in exactly the same way, transfer is total. In two experiments, where subjects perceived some type of difference between the old and a new application of a rule, transfer performance suffers. This was demonstrated in the two cases where newly generalized rules lead to longer learning times, which may be as long as original learning times. Since the additional learning time for one experiment was not as long as the learning time for new rules, it is hard to decide on the current evidence, whether generalization happens at all, and which the cases of generalization are. There is a chance that generalization will only occur if subject have learned the equivalence of two similar conditions declaratively. If this equivalence has to be induced during procedural learning, learning time is just as long as during new rule learning. Therefore it is possible that the level of transfer performance varies at least for productions low in the goal hierarchy with the level of perceived difference/similarity. What is perceived as similar or different may be subject to declarative representation and learning.

3.1.2 The Act* Research group.

As discussed above the transfer research of the Act* Research Group (Singley & Anderson, 1985; Singley & Anderson, 1988; Singley & Anderson, 1989) has a similar theoretical foundation as the research spawned by CCT. Again transfer predictions are made based on the number of new productions that have to be learned for a given task (see chapter 2 above). However, the methods of measuring and modeling transfer are different enough to make a direct comparison between experimental results difficult.

3.1.2.1 Experimental Procedure. Table 2 provides an overview of the experiments reported in Singley and Anderson 1989 and elsewhere. Anderson and his colleagues have collected transfer data in a wide range of domains, including complex problem solving tasks such as programming Lisp and solving Calculus problems. All experiments were monitored and controlled by computerized tutors using a training method similar to the one used by the CCT group. Usually, general instructions are given in a first lesson. Then the tasks are practiced by giving feedback after each error and providing help after two consecutive errors. In these experiments usually learn a wider range of sub tasks, and the learning is carried over longer intervals, sometimes over long sessions on several consecutive days. Furthermore, subjects do not learn to a given criterion defined by a series of error free trials. Instead subjects simply work on a set of problems for a given time, or subjects complete a set of problems where the goal is to finish the task (not necessary the error free handling of the task). One problem of direct comparison is therefore the argument, that different stages of original learning might have been achieved across paradigms.

The main differences exist in the measurement of transfer: In the Act* approach the units of measurement are usually some speed measurement for a given task unit (i.e. time per correct operator, planning time per subgoal, time to finish problem, etc.) and some type of accuracy measure (% of errors, keystrokes per trial, error per operator). Task units are defined according to varying levels of analysis. In a sense this might be a more noise free measurement of improvement, because polluting performance components (such as the keying rate) that do not improve across trials can be subtracted from units (such as planning time) that do matter.

Measurements such as these are used to calculate transfer using Gagne's transfer formula.

$$\% T = \frac{\text{train}(\text{day } 1) - \text{transf}(\text{day } n)}{\text{train}(\text{day } 1) - \text{train}(\text{day } n)} \quad (1).$$

Here, transfer is expressed as improvement due to transfer relative to improvement due to original learning. For example if a subject

Authors	Domain	Interface style	Transfer	Measurements	Type of modelling	Predicted Transfer	Observed Transfer	Fit to model
Singley and Anderson, 1985; 1987 (exp.1); 1989, chpt. 3	Text Editing	Command based line, screen editors	Within task, across editors, within style, across style	seconds/ correct operation, keystrokes/ trial, sec/keystroke planning time execution time	GOMS analysis, (p) transfer = $f(\text{known prod.}) / f(\text{total prod.})$ (o) transfer = $\frac{I(1) - t(n)}{I(1) - I(n)}$	LL ED EDT 68% EDT ED 75% Li Emacs 39% Ty Emacs 19% MT ED EDT 90% EDT ED 85% Li Emacs 27% Ty Emacs 7%	(plan time) LL 87% 91% 61% 35% MT 105% 99% 62% 29%	Surplus transfer for all units of comparison
Singley and Anderson, 1987 (exp.2); 1989, chpt.4	Text editing	Command based, screen editor	Within task Within domain, across editors, A-B, A-Br, max. interference Design	same as above	GOMS analysis (p) transfer as above (o) transfer = $\frac{I(1) - \text{trans}(n)}{I(1) \times r}$	LL EM pEM 75% MT EM pEM 85%	(plan time) 68% 63% (keystroke/ trial) LL -234% MT -61%	transfer under-predicted for both units of comparison negative transfer as transfer of nonoptimal methods
Conrad and Anderson, 1988; Singley and Anderson 1989, chpt 2	programming Lisp	Tutoring system	Within task within domain	time per production error per production, recorded by tutoring sys.	qualitative fit of learning and transfer curves to Act* learning curve	100% from one application of production to another	close to 100% within lesson, less than 100% across lesson due to forgetting (or warm-up)	good fit between predicted and observed levels of transfer
Kessler and Anderson, 1986; Singley and Anderson, 1989, chpt. 4	programming Lisp	Tutoring system	Transfer between iterative and recursive functions, between tasks	time to finish problems, verbal protocols	direction and amount of transfer is matched with data from verbal protocols (o) transfer = $\frac{I(1) - t(n)}{I(1) - L(n)}$		time to finish: iter -> recur 67% recur -> iter -20%	Verbal protocols show that negative transfer is due to transfer of nonoptimal learning strategy from recursive to iterative

Table 2: Summary of Transfer experiments specific to common productions hypothesis by the Act* group.

Authors	Domain	Interface style	Transfer	Measurements	Type of modelling	Predicted Transfer	Observed Transfer	Fit to model
Singley and Anderson, 1989, chpt. 5	Calculus problem solving	Tutoring system	Integration to differentiation Transfer between different wordproblem stories Selection of operators -> Application of Operators and vice versa.	time per operation % incorrect per operation extra moves (in addition to optimal)	(o) transfer = $\frac{l(1) - t(n)}{l(1) - l(n)}$ correlation of transfer for operator selection and application with the frequency of practice on these subtasks		100% transfer within translation part of task (no effect for stroy type) no transfer from or to translating to selection or application integration practice transfers to diff. selection 64% integration practice transfers to differentiation application 57%	Frequ. of selecting an operator (day 2) correlates with transfer (day 3) $r = .87 \times$ Frequ. of appl an operator on day 3 coreelates with transf. (day 3) $r = .48$
Kessler, 1988 Singley and Anderson, 1989, chpt. 5	Lisp programming	Tutor	within domain across task use specificity for coding debugging evaluation	timer per problem errors per problem	(o) transfer = $\frac{l(1) - t(n)}{l(1) - l(n)}$ no production system analysis	It is assumed that there is no overlap in productions across the task, therefore the transfer should be low across the board	Transfer based on time and error measure varies between 12% and 92%, with one fairly negative value (-66%)	Transfer is under-predicted by broad brush use specificity hypothesis.

Table 2 (continued).

finishes a task in 48 minutes during original training, in 38 minutes after being trained on a similar task for 3 days, and in 35 minutes due to training on this task for 4 days, transfer would be $48-38/48-35 = 77\%$. We will refer to this measurement as the 'percent saving' score. This percent savings score is then modeled or explained in various ways across experiments: In some cases these scores are simply compared to gross qualitative hypotheses about transfer based on Act* (Conrad & Anderson, 1988, and Kessler, 1988). In other cases, converging evidence for explanations of the magnitudes of transfer is collected from verbal protocols (see Kessler & Anderson 1988), or by correlating other data (such as frequency of practice) with the transfer scores (Kessler & Anderson, 1988 and Singley & Anderson, 1987). In one series of experiments (Singley & Anderson 1987) percent saving scores are compared directly to quantitative transfer predictions derived from a production system analysis of the given tasks. These transfer predictions are derived by

$$\%T = \frac{\text{frequency (known - productions)}}{\text{frequency (total - needed - productions)}} \quad (2).$$

If for example a subject is believed to have learned 14 relevant productions, but 36 additional productions are needed to execute the task, the subject would be expected to show 28% transfer provided all productions are only applied once for this particular task. To correct for the frequency of their use, the number of known and needed productions is multiplied with their application frequency. Here is another difference between the CCT and the Act* modeling approaches: CCT bases its transfer predictions on the raw number of new rules, whereas the Act* approach corrects for application frequency. In some situations, this formula might under predict transfer by overemphasizing the weight of new rules. New rules should only counted as 'new' for their very first times of application (see the results reviewed above). If a new rule is used many times during a given task unit, then formula (2) above can inflate the nominator unduly and subsequently under predict the amount of transfer that can be expected.

3.1.2.2 Experimental Results. Taken as a whole, the data summarized in Table 2 provide relatively good support for a common productions theory of skill transfer. In the following section, we will take a somewhat superficial look at three experiments (Conrad & Anderson, 1988, Singley & Anderson, 1985, and 1988; and Singley & Anderson, 1989, chpt. 5). On a general level of analysis these experiments show perfect transfer from one application of a set of productions rules to the next, satisfactory correlation between predicted and observed transfer (in terms of formulas (1) and (2) above) between different sets of operators, as well as evidence for the fact that in one domain the transferred productions must be fairly general. These results generally support the view of productions as the basis for transfer.

In the next section, we will review some of the same experiments in some more detail. At this level of analysis it becomes clear that even though common productions may be a good approximation for predicting transfer, there is evidence that transfer has been repeatedly under predicted, we call this phenomenon 'surplus transfer'. Let's first begin with the general analysis, however.

Self Transfer. Conrad and Anderson, 1988, using the domain of programming found total transfer from practicing a set of nine new productions of from one practicing trial to the next. One day delay between practicing trials, slows the time per production somewhat (probably due to forgetting), but this slowdown is overcome in a second trial after delay. The data, which are collected over several applications of the same set of productions within sessions and across sessions do not suggest that warm-up plays a big role for production execution time. Time decreases somewhat after the first trial of a given session, but the subsequent trials stay at about the same level.

Transfer to a similar task. In the domain of text editing (Singley & Anderson, 1985, 1987, and 1989 chpt. 5), it was demonstrated that levels of predicted transfer correlate fairly close with levels of observed transfer, for measures of planning time and time per correct operation. In one study (Singley & Anderson, 1987, exp.1), planning times associated with particular subgoals of the task showed similar levels of transfer as calculated from formula (1), when compared to predicted levels of transfer derived from formula (3). Even though predicted and observed levels of transfer was well correlated, observed transfer was generally somewhat under predicted (see Table 2). We will discuss the second result in detail below. A second experiment in this study (Singley & Anderson, 1987, exp. 2) also showed high levels of transfer in the planning time component, that are nevertheless slightly under predicted. For this experiments , transfer was present even for planning time on low level commands that had been rearranged to fulfill the maximal interference design in the sense of the verbal learners (A-B, A-Br), even though a little less than predicted. This seems to be further evidence (see Polson, et al., 1987 above) to suggest that re-pairing of known responses does not lead to interference. However, a look at the number of keystrokes as a measure of accuracy and optimal use of the commands, shows strong negative transfer. Further analysis of the type of commands chosen in the transfer task shows that this negative transfer can be explained by transfer of non-optimal methods from the practice task, as well as a tendency to choosing commands who have not been bound in the previous task (and that might have been easier to learn therefore).

In a third experiment (Singley & Anderson, 1989, chpt.5) it was shown that transfer within task components occurs even across task context for the domain of calculus problems . The investigated components were (1) the translation of word problems into equations, (2) selection of operators, given a problem statement and a goal, and (3) the application of operators for

calculus problems. If only selection was practiced, the subject would simply choose one of seven operators (such as 'differentiate', or 'restate'), and the tutor would then apply the chosen operator to the given equation. In the application training condition, the tutor chose the operator, and the subject practiced applying it to the equation. Even though these subskills were practiced in a task contexts different from the contexts of the transfer tasks, (different story types for the translation component, and selection and application of operators in either differentiation or integration problems), the transfer scores were high and relatively stable.

3.1.2.3 Summary.

Overall, these experiments seem to demonstrate that (1) transfer across and within tasks measured as percent savings is correlated to the proportion of known productions; (2) negative transfer can only be observed at a very low level of analysis and is mainly an effect of transfer of non-optimal methods; and (3) that at least for the domain of calculus, productions in the sub skills of translation, operator selection and application, must be represented in a fairly abstract way, because they transfer across different task contexts. This pattern of results seems to extend and confirm the results from the CCT research group and could lead to a full support of a common elements theory of transfer.

We will now take a second pass through these experiments, (Singley & Anderson, 1985, 1988, and 1989, chpt.5), and show that even though the general fit of the data to the models are satisfactory, unpredicted levels of transfer are found in a range of measures. The analysis will also show that most of the unpredicted transfer is found in experiments that were designed to test the use specificity hypothesis. We will start with the description of surplus transfer in the text editing experiment.

3.2 *The phenomenon of surplus Transfer.*

3.2.1 Surplus Transfer in Text Editing. As noted above, comparisons of the general directions of predicted and observed transfer scores of three experiments show a relatively stable amount of under prediction of transfer (Singley & Anderson, 1987, Singley & Anderson, 1989, chpt.5, and Kessler, 1988). The combination of these two results, a relatively good fit of the common elements model, but some under prediction of the amount of transfer suggests, that the common element model only tells part of the story about transfer. Some of the experiments seem to allow for transfer based on knowledge that was not captured by analyses based on common productions. To collect some hypotheses about the locus of this effect, we will now describe the experiments that demonstrated it in detail.

In the text editing experiments (Singley & Anderson, 1987), transfer between a range of different text editors was investigated: subjects were transferred from one line editor to another, from practice on line editors to a screen editor, and a control group was transferred from typing to using the screen editor. Subjects received marked-up copies of manuscripts, in which they had to perform edits such as deleting and inserting words, characters, or lines. All four tasks (editing in the two line-editors, the screen editor, and the typing task) were analyzed according to a GOMS analysis. The top level goal structure of this task decomposition was the same for the three different text editors. It included subgoals such as 'acquire the task unit', that is the reading of the manuscript and extracting the instruction for the next modification of the text. The next subgoal is to 'locate the line' in the document that contains the error (LL), and the final high-level subgoal is to modify the text (MT) after it has been located.

The subjects' keystroke sequences and pauses in the two main sub tasks, LL (which includes the acquisition of the edit) and MT were analyzed with the help of a parser. Longer pauses between keystrokes were counted as planning time, bursts of keystrokes were accumulated into executions time, so that both components of time could be separated for LL and MT. The values of the observed transfer scores for the planning component based on these analyses and formula (1) are included in Table 2. The table also includes predicted transfer scores based on the count of common productions and formula (2). The comparison of these values demonstrates clearly that all the observed transfer scores are above their predicted counter values.

Singley and Anderson have been trying to explain this surplus transfer in two ways: (1) One attempt was to break the analysis of the MT transfer scores down further into planning times associated with single edits (such as 'delete word', or 'insert character'). The analysis on this level showed that planning times for delete operations was connected to the highest and lowest (negative) transfer scores, for both subjects transferring from the typing controls, as well as from the line editors. This result was further explained by showing that all subjects (even the typing control group) had some practice in using the delete key, and that this practice transferred positively where the new editing method was similar, and negatively, where repeated use of the delete key less effective than the use of the actual command sequence (for strings and lines). Unfortunately, this story does not explain transfer for the other commands. Transfer for insert and replace operations varies between -39 to 74% for the typing control subjects, and between 33 to 77% for the line editor subjects. No special function keys were assigned to these operations that could have been practiced by the typing subjects before. The relatively high transfer scores for these operators therefore remain mysterious.

(2) A second explanation involves practice and transfer based on the 'acquire unit' task. As mentioned above, this involves glancing at the

manuscript and encoding the type of edit, as well as the particular changes to the original text. An explanation of transfer based on this component involves improved encoding skill, both for the type as well as for the content. This would explain subjects performance in all transfer conditions, because all subjects had equal amounts of practice in reading manuscripts and interpreting them. Singley & Anderson report anecdotally, that subjects glanced at the manuscript several times, even while performing the MT sub task. The parser however could not distinguish these glances from planning times associated with other cognitive operators in either subgoal of the task (MT or LL). The improvement on this skill component could therefore explain the additional transfer in the planning times, observed both for MT and LL. Additional evidence for this interpretation can be taken from the performance analyses of Bovair, et al. (1990) reported in the second chapter. Here, a multiple regression to explain performance times on a similar text editing task, showed that a predictor based on the number of editing feedback checks explained a significant amount of variance. Moreover, there was a significant interaction of this parameter with the level of expertise of the subject. If subjects were fairly skilled in using the word processor, they used significantly less time checking the results of their actions than if they were novices. We can conclude then, that surplus levels of transfer in the current experiment may be related to this dropout of additional glancing at the manuscript, either for reasons of improved encoding skill, or because subjects simply did not check the results of their actions as frequently.

The most interesting results concerning surplus levels of transfer were accumulated in experiments that concerned the hypothesis of use specificity of knowledge. What differentiates these experiments from the ones reported so far is the minimal overlap of shared productions from one task to another. However, in these experiments different sub skills in one domain are based on a large common declarative knowledge base.

3.2.2 Use specificity? All other studies that have yielded unpredicted levels of transfer were designed to show that transfer can be explained only by use of the same productions, *not* by shared general knowledge. Anderson calls this the use-specificity hypothesis. For example learning to read a foreign language should not transfer to learning to speak the same language, even though the same declarative knowledge (vocabulary, rules of grammar) might be involved. Curiously all experiments designed to demonstrate this effect have failed, by displaying much larger transfer effects than could be explained on grounds of a common productions theory alone.

3.2.2.1 Transfer in Lisp Programming (1). Kessler (1988) performed an experiment to show that different sub skills in the domain of Lisp programming (coding, debugging, and evaluation) do not transfer to each other. Subjects were trained in one of the sub skills and then transferred to one of the other conditions. Percent savings scores based on times to finish a problem and errors per problem show considerable levels of transfer (between

12% and 92%, with only one negative value (-66%). The highest levels of transfer were observed between the coding and debugging sub skills, which lead Kessler to hypothesize that common productions between these two skills might be responsible for this result. A more detailed analysis of the data (based on a post hoc production analysis), comparing the time and error measurements for the common and unique components separately, did not support this hypothesis however. Even though the transfer scores were higher for the common productions, the savings scores for the unique components was still considerable (an average of 52 percent). In discussing this result, Singley & Anderson list as one of the possible candidate transfer component the improved encoding of the problem statement (Singley & Anderson, 1989, p. 162).

3.2.2.2 Transfer in Calculus (1). In this study, Singley and Anderson (1989, chpt. 5) tried to demonstrate the effect of use specificity in the domain of calculus. With respect to the subskill factor, the experiment had three conditions (translation of a word problem into an equation, selection of operators to apply to that equation, and the application of these operators). The transfer task for all subjects was solving differentiation problems under the use of all components. For each subskill, transfer was measured in terms of speed, such as time per operation, and measures of accuracy, such as % incorrect. Subjects received different levels of training for the different sub-components. On the first day of the experiment, all subjects were introduced to the new operators and practiced application of the operators on differentiation problems. On the second day of the experiment, the participants were subjected to differential treatment: One group of subjects practiced all subskills in the context of differentiation problems (the control group). One other group practiced all sub skills in the context of integration problems. The third group finally, practiced application of operators in the domain of differentiation problems, they did not have to translate, or select the appropriate operators, however.

Since this experiment does not separate the amounts of practice in the three components neatly from each other, we have to be careful in interpreting the results. Most of the calculated transfer scores (time and % incorrect, respective extra moves per problem) for the translation and selection sub-component are calculated related to performance in these components on day 2. However, this does not take into account that the application practice that subjects received on day one may have already influenced their 'novice' scores on day two on those measures.

For example, the translation times and % incorrect measure for subjects that did not practice translation on the second day are at the same level as the translation times and % error for the subjects who first translated problems on day 2. However, all of these subjects had already worked on applying the operators to equations on the first day of instruction, and had therefore some exposure to the type of equations that they were expected to find. Therefore the only conclusion from this part of the data can be that a

second day of practice applying operators does not improve the translation speed and accuracy, but we simply do not know whether one day of practice applying operators makes any difference opposed to no practice on these types of calculus problems.

The same caution has to be taken with regard to the transfer measures for the selection sub component. Again, the time and error measures of the group that did not select operators on day two (but applied them) are very similar to the selection measures of those groups that practiced all sub components on day two of the experiment. However, all of the subjects received some practice in applying operators on day one, and it is well possible that this practice had some effect on selecting operators on the second day. Again, the only conclusion to be drawn from the data is that a second day of practice applying operators did not improve the selection scores over and above their levels after one day of practice.

Singley and Anderson were successful in relating different levels of savings for performance on selection and application of particular operators to the frequency with which these functions had been practiced on day two of the experiment. The correlation coefficient for frequency of selection and percent saving on the selection sub component on day three is $r=.87$. Even though this correlation suggests a relatively smooth relation between frequency of practice and transfer for selection, one has to keep in mind that the frequencies of selection and application of an operator are perfectly confounded on day two. In other words, the same correlative relationship would hold between frequency of operator application (on day two) and transfer on selection on day three. It is therefore impossible to tell if it is indeed practice on the same sub skill or the 'unrelated' sub skill that lead to the difference in transfer scores.

The same criticism holds for the interpretation of the savings scores for the application component. Again, frequency of practice of applying certain operators on day two can be correlated with percent savings on application of those operators on day three ($r=.54$). Again, the frequency of application for a given operator on day two is confounded with the frequency of its selection. Therefore the same correlative relationship also holds for practice on selection and savings scores for application. Again we can only conclude that operators that were practiced more frequently on day two (in selection and application) were applied faster and more error free on day three or the experiment.

To summarize the results: First, a second day of practicing application of operators does not improve selection or translation performance overall, however the frequency of practice on selection and application predicts the level of performance on either component on a subsequent day relatively well. These results leave obscure whether a single day of practice on application would have transferred to either sub skill, or whether practice on selection or application predicted performance on either sub skill on day

three. To answer these questions, Singley and Anderson (1989, chapter 7) designed a second experiment. In this experiment transfer was measured for one day of practice on the same of the other sub skill.

3.2.2.3 Transfer in Calculus (2) In a follow-up to the calculus experiment reported above, Singley and Anderson (1989, chpt. 7), designed an experiment to test specifically, whether practice on one component (application or selection) of the skill would improve initial learning on the second. This second experiment included two experimental and two control groups. The two experimental groups practiced selections or they practiced applying operators, after the tutor had selected them for them. A baseline control group practiced typing selection and applications without actually solving the problems, another group of subjects practiced solving both parts of the problems.

Performance data on day two, where all groups select and apply operators were analyzed. The selection and the application sub skills were analyzed individually.

Both, practice on the selection component, as well as practice on the application component, lead to significant increases in speed of selection, whereas the selection speed of the typing control group did not differ from the novice group performance on day 1. These differences do not replicate for the 'extra moves per problem' measurement, which can be interpreted to indicate that none of the groups really learned more efficient selection heuristics.

For the application sub skill, speed of application on day two is dependent on application practice. The effect of selection practice on speed of application is not statistically significant, even though there is a numerical difference between subjects who practiced selection and those who did not. Again there is no improvement in the accuracy measurement (percentage of incorrect applications) due to practice on any of the sub skills.

In summarizing the results from these two calculus experiments we can now conclude, that first, one day of practice applying operators transfers to selecting operators, even though a second day of operator practice does not seem lead to further improvement. Second, improvement on the selection component is mainly due to increased speed of selection, not to better selection strategies. Here is our first hard evidence for transfer in spite of a lack in overlap between productions in the sub skills. Interpreting the fact that only the speed measure, but not the efficiency measure shows improvement, we could suspect, that this improvement could be due to faster access to the set of operators, whereas subject have not learned yet, which operators will be optimal to choose for a given problem statement. Thirdly, practicing selection of operators does not seem to have any beneficial effect on application of operators, even though there is improvement due to practice on application. This result has been explained by suggesting that practicing

operator application improved the declarative representation of the operators. The faster selection times for subjects on the second day were then a consequence of access to this improved representation. However, the processes involved in selecting operators did not elaborate declarative knowledge in a manner that the more complicated processes involved in applying operators could benefit from it.

In the following experiment where transfer between different sub skills in the domain of Lisp programming were investigated, we will find more evidence for the effect of transfer due to an improved declarative representation of the domain.

3.2.2.4 Transfer in Lisp programming (2). Another recent experiment, designed to test the use-specificity hypothesis, shows large transfer effects that can not be accounted for by a common elements theory based on production overlap (Pennington and Nicolich, in press). The experiment tested transfer between generation and evaluation of lisp code, in extension of an earlier study by KcKendree and Anderson (1988). Transfer was measured in both directions, and the experiment included training control, as well as typing control groups. Subjects in the experimental groups practiced one sub task for three days (two hours per session), and were introduced to the transfer task on day four. Subjects in the training control groups simply practiced one sub skill for four days, and typing subjects practiced typing either evaluation or generation problems before they were transferred to one of the problem solving conditions.

The generate and evaluate sub tasks were analyzed and two production system models to simulate task performance were constructed. Comparison of the productions between these two models revealed that there was very little production overlap between the two skills. This overlap consisted of six productions concerned with parsing of lists. Predicted transfer (under application of formula (2) from generate to evaluate and vice versa was 22% and 12%, respectively. Learning and transfer was measured both for speed (plan time and number of pauses) and accuracy (% correct, and error/stroke ratio), as well as in a combined speed/accuracy score. From these raw performance measures percent savings scores were calculated under use of formula (1). The results of this experiment are summarized in Table 3.

Comparison of these values with each other and the predicted values, illustrates the following points: First, overall transfer is higher than predicted (an observed average of 55.5% as opposed to a predicted average of only 17%). Obviously, the common productions model under predicts transfer seriously in this study. Second, transfer was predicted to be higher for generate to evaluate, but the observed savings scores are opposite to that (with one exception: plan time). This is the first time that transfer scores are not only higher than predicted, but also poorly correlated with the direction of transfer. Third, there is transfer for all measures, speed as well as accuracy indicators. Subjects made faster decisions as to what to do next (as expressed in plan time

and pauses), and they made more efficient decisions (perc. correct, and error/strokes). In this case subjects did not only have faster access to operators they were also able to make more accurate decisions from the very beginning as an effect of transfer.

Measurement	Transfer direction	
	generate-evaluate	evaluate-generate
perc. correct	40.1%	61.0%
error/strokes	60.5%	88.2%
pauses	51.6%	47.9%
plan time	54.1%	61.0%
points	34.1%	48.3%
average	49.9%	60.7%
gross average	55.3%	

Table 3. Summary of transfer results from Pennington and Nicolich (in press).

Principally, there are three alternate conclusions that could be drawn from this study. First, it could be argued that the production system analysis of the data was flawed, and that a different set of productions could be written that predicted the observed transfer better. A second alternative is that the analyses provided were correct, but only described performance on the sub tasks when they were learned by pure novices. There is a possibility that transfer subjects adopted altogether different problem solving strategies when transferring to the other sub task, and this transfer performance would then be ill described by the original performance model. Thirdly, one could decide to take the results at face value and accept the fact, that in this transfer can not simply be described by a common productions model. This hypothesis leaves us with the interesting duty to further investigate what the basis for this type of transfer might be. This was done in a follow-up experiment by Pennington, Nicolich, and Rahm, (submitted). We will describe this study in detail after summarizing the results that we have accumulated so far.

3.2.3 Summary. What are the commonalties and differences then between these experiments additional to their common overall result of surplus transfer?

As mentioned above, four out of four experiments designed to demonstrate use specificity, but only one out of ten experiments to show transfer between similar tasks within a common general domain resulted in surplus levels of transfer. The main characteristic of the use specificity experiments is that there is a clearly shared declarative base between the sub tasks. This declarative knowledge includes the familiarity with a set of new lisp terms, mathematical operators, the ability to parse lists into its elements,

and to encode equations efficiently, etc.). However, there is no or very little overlap in terms of the sets of productions that describe skilled cognitive processes in either sub tasks. In case of the editing experiments, there is some overlap in the declarative knowledge needed for the given tasks (basic facts about the devices, and basic knowledge of the task, for example what it means to perform an edit such as inserting or replacing etc.). However, the same knowledge is also represented in production rules that are shared between the separate tasks for the experiment. They can therefore transfer *as productions* from one application to another. If new rules have to be learned for a given transfer task in these experiments, there is no overlapping declarative knowledge that could assist in their acquisition. For example, in the transfer tasks, new command sequences are introduced, that simply have to be learned from scratch. We also have evidence from the CCT experiments that learning to generalize a rule to a new situation comes with a cost: Subjects needed extra time to recognize that an old rule (for deleting a document for example) applied whether the menu choice was called 'delete' or 'discard'. In this case it would be easy to imagine that previous declarative instruction about the equivalence of delete and discard in the context would have led to transfer of these productions.

To summarize this point: There might be a learning strategy by which a new problem is first attempted to be solved under the use of well-known productions. If this strategy breaks down, subjects start searching for facts about the domain that are well known, and general problem solving mechanisms are applied to these facts. If these declarative facts are new and poorly integrated into the subjects general knowledge (as a set of calculus operators, or knowledge about lisp expressions and lists would probably be), then elaboration during first task learning may have a beneficial effect on learning a second task that is also based on the same declarative knowledge base.

There are two further experiments that investigate this hypothesis in more detail. One study manipulated declarative learning directly and assessed its effect on subsequent procedural learning in using some of the learned facts. This experiment will be described next. Finally, the follow-up to the Pennington and Nicolich Lisp study will be reported, in which it was ruled out, that a flawed task analysis was responsible for the unexplained transfer effects.

3.3 *The case of declarative transfer.*

3.3.1 Declarative Transfer in operator selection. For the domain of Logic theorem proving, Lewis and Anderson (Singley & Anderson, 1989, chpt7.) conducted an experiment to investigate the effect of declarative training on procedural transfer more directly. In this study, subjects either practiced recognizing a set of logical rules from a set of foils for two days (the declarative learning condition) or they received practice in selecting the rules

to prove theorems for the same amount of time (the procedural learning condition). On the third day, half of the subjects were either transferred to the alternate task or received a third day of practice on the original task. With this design, the effect of declarative practice on procedural learning and vice versa could be investigated. The results were the following. First, time per rule selection on day three was clearly reduced by prior practice on rule recognition. In other words, the declarative practice had a beneficial effect on the speed of executing the procedural task. On the other hand, practice on rule selection did not transfer to rule recognition latencies. Practice on the procedural task therefore did not improve performance on the declarative task. It seems that whatever is practiced in rule selection does not improve the declarative accessibility of the given rule set. Singley and Anderson interpret this as evidence for the procedural manner of the practice, which in turn leaves the declarative traces of the rules untouched. This argument however stands in contradiction to their interpretation of the transfer between application and selection components in the calculus experiments: there we argued, that practicing application of the operators (a *procedural* task) elaborated or practiced the declarative representation of those operators, which lead to increased performance on selection. A more consistent argument might be that performing the rule recognition task in the logic experiment requires a detailed level of representation of the rules (specific knowledge of surface elements) that is more than sufficient for the rule selection task. Practicing selection did not lead to elaboration of the same level of detail. Therefore performance on the recognition task did not improve after practice on rule selection. We can therefore conclude that deliberate declarative practice does produce transfer results for simple procedural tasks, such as selection of logic rules and possibly selection of calculus operators.

3.3.2 Declarative transfer in sub skills of Lisp programming. In a recent follow-up experiment, Pennington, Nicolich and Rahm (submitted), investigated the source of the large transfer effects they had established in the earlier Lisp experiment further (Pennington & Nicolich, in press, see above). In this experiment the identical training paradigm as in the earlier study is used to train four subjects on either evaluation or generation of lisp code. After three days of practice the subjects are transferred to the other sub task. These subjects were videotaped and verbal protocols were collected in addition to the performance data. The goal of this study was twofold: In a first set of data analyses, the two hypotheses that suggest an error in the task analysis (either in general, or more specifically for the transfer performance) were investigated. The second part of the data analysis tracks the development of the declarative knowledge base during initial procedural learning of either sub task.

For the first set of analyses the subjects videotaped problem solving behavior as well as the verbal protocols from two problem solving sessions during the third day of practice and day four (the transfer task) were scored for the use of a set of pre-defined mental operators (the observed data set). These

mental operators were derived from the production system models that had been used to predict transfer performance. These production systems were run on the same tasks. From the model runs the number and sequence of operator applications were abstracted (the predicted data set). Two types of scores were calculated to evaluate whether the number and sequence of operators executed by the model were well matched by the number and sequence of operators chosen by the subjects: First, model and behavior fit and a concordance score were derived. These scores express whether the number and the type of operators produced by the model and the subjects matched well. The scores indicate that the model does not only describe practicing data (day 3 performance) but also transfer data (day 4 performance) extremely well. All the derived values are over 90%, for both sub tasks.

Second, to further evaluate the model predictions in terms of the correct sequencing of operators, a second set of analyses were performed. For these a proportion of the number of sequences that overlap between two sets of sequences were calculated, correcting for the number of possible sequences of a certain size, as well for their overlap by chance (ARC scores). These scores again show an excellent fit between the generation and evaluation model's predictions and the observed behavior, for both training as well as transfer performance (with ARC scores higher than .67 for sequence sizes of three and four). We can conclude the production systems used in this experiment did not only predict the type of operators used by subject, but even the correct sequencing during application. These analyses then rule out that a flawed task analysis was responsible for the discrepancy between predicted and observed transfer scores in the earlier experiment.

For the second set of analyses, the results from a declarative knowledge test, given at the beginning of the training phase, and the verbal protocols from the first learning session (day 1) were scored for evidence of known or inferred declarative knowledge elements. Declarative Knowledge was represented as a network of nodes with directed links. Concepts such as 'list', 'element', 'first' or 'insert', as well as well as related attributes such as 'returns first element of a list' were represented as nodes. Whenever a subject openly mentioned one concept after considering another, or whenever an action implied, that such knowledge must have been accessed, a link was drawn between the respective nodes. In this manner, correct, as well as incorrect inferences were recorded. For each trial the previous representation was updated. Updating was established by incrementing link strengths for each correct access of that trace. If new concepts of relations were mentioned, they were simply added to the old representation. This process produced 60 representations for each subject, where each representation stood for the subject's declarative knowledge after a certain number of practicing trials.

Comparison between the final representations provides us with the following information: First, subjects elaborate their initial declarative knowledge structure (from reading instructional texts) a great deal by applying it to practice problems and interpreting feedback from the system. Secondly,

the complexity of the knowledge maps increases greatly during the first 30 practice trials, but does not extensively extend thereafter. Third, the final knowledge maps derived from subjects practicing generation and evaluation contain extremely similar nodes and links. However, despite their general similarity, the maps derived from the protocols of the generation practice subjects are somewhat more complex. This added complexity represents some wrong inferences that had been drawn by subjects in this condition as a consequence of false interpretations of error feedback. This analysis therefore shows that the declarative representation of the domain is very similar, no matter if it was established during evaluation or generation of the Lisp problems. This shared set of declarative knowledge therefore is a very likely candidate to explain the high transfer scores in this and the previous experiment. In fact, the slightly less wrong representation derived from the generation subjects might be an explanation for the finding, that subjects that transferred from evaluation to generation did somewhat better than their counterparts.

In this study then we have further unanimous support for the hypothesis that transfer of learning can happen on the back of a well practiced and elaborated declarative knowledge base, even though only minimal overlap in terms of productions exists. The study provides strong evidence for the hypothesis that high transfer in this and the earlier experiment were due to use of the practiced (incrementation of 'correct' links) and elaborated (learning of new concepts and links) declarative representation of the skill. This declarative learning took effect after the 'official' declarative learning phase (the reading of the instruction booklet). The analysis of the verbal protocols clearly showed that subjects actively enriched their representation of the new concepts during problem solving, through the processes of self explanations and the interpretation of feedback. This study leaves us with the important conclusion that declarative learning of new concepts, operators and symbols (such as the parsing of lists and their elements), is a slow and gradual process. This process may overlap with the initial phases of procedural learning in the domain, in fact the process of procedural learning may be instrumental for the acquisition of declarative knowledge. Whenever such declarative learning happens during the initial practice trials, transfer is possible in terms of access to a richer and better practiced declarative representation of the new domain.

3.3.3 Summary. Both, the logic theorem proving experiment, as well as the Lisp experiment provide evidence for the importance of declarative transfer. It is still an open question what type of declarative learning (and what type of representation) will improve procedural learning most. The former experiments (the logic and calculus studies) suggest that simple familiarization with the new concepts merely improves the speed of access to those concepts, but not the accuracy of the selection of operators. This result suggests that this type of learning did not improve differentiation between the new concepts. In the Lisp experiment, subjects did not only learn the recognize the concepts, but they explicitly learned their distinctions, and the

contexts in which they applied. An error analysis (that we did not report above) suggests, that a great amount of error reductions was due to better differentiation between concepts that could be easily confused. This suggests, that in order to reduce errors in initial procedural learning the concepts have to be well differentiated.

We will discuss in the next chapter that these two components of declarative learning, familiarization with a set of new concepts, as well as differentiation between them, have been the focus of investigation in the transfer work of the verbal learners. First, the experimental results reviewed in this section will be summarized.

3.4 Conclusions.

There are four general conclusions to be drawn from this survey of transfer research. The first concerns the conditions under which a common productions theory can explain experimental data well, the second concerns the possibility that even in those cases transfer might have been due to declarative transfer. The third issue concerns the effect for theories of learning and transfer. A final issue addresses the proper level of analysis for investigating transfer and leads into the final chapter of this paper.

(A) Common production theories make excellent predictions about observed levels of transfer in certain conditions: First, if there is a fair amount of shared knowledge between skills in terms of productions, and new productions have to be acquired from scratch (without benefit of access to practiced declarative representations). This was demonstrated for the case of the studies investigating transfer in the domain of software application skill. For these tasks all the relevant declarative knowledge practiced in the training tasks had also been incorporated in productions. If new productions had to be acquired in the transfer tasks the relevant declarative knowledge had not been practiced before (such as different command bindings etc.) Therefore declarative transfer did not play an important role in the acquisition of these new rules, and transfer could be explained perfectly on the background of shared productions.

Second, if the declarative concepts that are accessed during the first stage of skill acquisition are well known to subjects, so that additional processing of this information does not further change the accessibility and discriminability of this knowledge between the source and the transfer task. When these two conditions (large amount of shared procedural knowledge and general familiarity of the concepts in the new domain) are not fulfilled in an experiment (such as in the calculus, logic theorem and Lisp experiments), we can expect to observe declarative transfer.

(B) Even though common production theories explain transfer in the given conditions well, there is the logical possibility that what has been interpreted as procedural transfer in those cases, is in effect also the product of

declarative transfer. In the software skill experiments the overlap in terms of procedural and declarative knowledge between the source and transfer tasks might be perfectly confounded. Therefore transfer could probably be predicted based on analyses of both knowledge components. An experimental set up to investigate this possibility might involve training subjects to different degrees before transfer. In one condition subjects might be practiced to a degree after which the assessment of declarative learning does not show any further improvement (in the Lisp experiment after the 30st trial). A second group of subjects receives further procedural practice (performance will still improve further in terms of decreased planning time, and accuracy measures). If transfer on the basis of formed productions indeed exceeds transfer on the basis of a shared declarative base, the transfer scores of the two groups should be different from each other. A further clarifying manipulation would involve the number of learned and transferred productions by keeping the amount of declarative practice constant. If transfer depends on the number of shared productions, transfer should vary with such manipulations.

(C) Learning theories like Act* have underrated the importance of a declarative learning phase that may precede or overlap with the problem solving phase leading to compilation of productions. These theories have put most emphasis on the difficulty of learning the association between a given cognitive representation of a problem state and an appropriate operator to apply in this state. However, for many tasks the learning of such a connection might be relatively simple as compared to the acquisition of representations for a new set of concepts and cognitive operations. For example, new mathematical symbols or Lisp terms and code must initially appear as arbitrary meaningless characters to subjects. Depending on the number and similarity of such newly introduced concepts, it might be very difficult to encode and distinguish them successfully. The initial phases of problem solving may therefore be less concerned with learning appropriate condition - action associations as with learning about the involved concepts and operations themselves. However, there is the possibility that concepts will be defined by the types of actions that are successful in their context, or new actions may be distinguished by certain classes of conditions in which they apply. More research is needed that addresses the interaction between problem solving, concept acquisition and the acquisition of productions in the early stages of skill acquisition.

(D) This discussion illustrates that procedural transfer may have been analyzed on too high a level. We started with an analysis of transfer on the basis of relatively large knowledge units. Productions are relatively complex representations of cognitive states and cognitive operations. The importance of declarative practice for skill acquisition emphasizes that the acquisition of concepts that make up the conditions and actions of productions may precede their association and transfer independently. This hypothesis is paralleled by the development that was taken in the Verbal learning paradigm. In this earlier line of research, the initial concern was with investigation of transfer (or interference) based on Stimulus-Response associations. However,

evidence accumulated that suggested that smaller components of learning (such as stimulus and response learning and differentiation) were important factors in transfer in certain experimental conditions.

In the final chapter this experimental evidence will be reviewed to gather some additional information about the component processes that underly declarative learning and transfer.

Chapter 4. Transfer in the Verbal Learning Paradigm

There are two striking parallels between the directions in which research has been developing for both the verbal learners and the 'procedural learners'.

Both schools started investigating transfer from the absolute standpoint of explaining transfer in terms of a single phenomenon. The verbal learners investigated transfer in terms of associations between verbal stimulus-response material. We will try to show below that these associations might be described as declarative knowledge of concepts and their associations. In the process of investigating this phenomenon, interfering components of transfer were discovered. The first group ('general components'), which included warm up and learning to learn can be reinterpreted to be a true procedural skill component (see our arguments below). In the process of investigating what we might call declarative transfer today, these researchers therefore discovered what we might call procedural transfer.

The parallel to the development in the recent transfer work is that these researches started their investigation with the goal of exploring procedural transfer, but were surprised with high levels of declarative transfer. Both components therefore seem to be of importance.

The second parallel is with the level of analysis. Within the 'declarative' or 'specific transfer component' of the verbal learners, smaller components of transfer were discovered to be of importance than the learning of whole S-R units. In particular, the learning of a new set of stimuli and responses and the differentiation between them was discovered to be of great importance in certain experimental conditions. We suspect that the same decomposition of the declarative learning phase (into gradual concept learning and differentiation) also holds for the declarative learning phase connected to skill acquisition.

In the following section the work of the verbal learners will be introduced in general, and the components of transfer will be discussed in some detail.

4.1 General introduction

The verbal learners were essentially the predecessor to modern cognitive psychology. It was a functionalist approach to describe learning results in a stimulus-response paradigm, without any use of such mentalist constructs like memory, cognitive processes, representations etc., influenced by the behaviorist philosophy of the time. The goal was to establish functional laws that describe the relation between (objectively observable) stimulus and learning conditions and (objectively observable) learning

results. The verbal learners conceptualized learning as the establishment of associations between stimuli (word lists) and responses (particular words or meaningless syllables). Transfer was investigated mainly in its form of negative transfer or interference between previously established and currently learned associations between stimuli and responses. For example, negative transfer would be expected when a new response to a stimulus, that is already associated to another response, has to be learned. This would be the familiar phenomenon, in which a woman who changed her name with her marriage, will still be remembered by her prior name.

The transfer work produced by the verbal learners used exclusively verbal material in form of word lists, lists of meaningless syllables, or simple drawings of meaningful or meaningless patterns. In a typical experiment, a subject learns *paired associates*, that means they learn to associate a given list of stimuli (for example a list of words) with a given list of responses (for example a list of meaningless syllables). Let the notation A-B stand for a certain stimulus A paired with a certain response B. After learning a list of such paired associates to a certain criterion, the subject then is transferred to the learning of a second list. For example, a list A-B has to be learned in the original trial and a new list A-C has to be acquired in the transfer trial. This means that the original stimuli are paired with a new set of responses in the transfer condition. In this case moderate negative transfer is expected in the learning of the transfer task, because previously learned associations *interfere* with the acquisition and retention of the new responses. Table 4 summarizes the five basic transfer conditions (Postman, 1971), and the predicted transfer results based on the establishment of S-R associations and interference alone.

Design	Description	expected Transfer
A-B	A-B stimuli and resp. stay same	positive
A-B	A-C same stimuli., new resp.	moderately negative
A-B	C-D control, all new material	zero
A-B	C-B new stimuli, old responses	positive
A-B	A-Br new combination, old material	strongly negative

Table 4. Basic Transfer Designs of the Verbal Learners

As Martin (1965) summarized, the early work of the verbal learners (i.e. Müller & Schumann, 1894, and culminating in Osgood's (1949) transfer surface) saw positive or negative transfer mainly in terms of transferring or interfering associations between certain S-R combinations. However, researchers in the 1960's and 70's had to recognize that learning and transfer

times are an additive of a range of components, such as learning to learn, warm up, response learning and so on (see Table 5 below). In fact, research during these years was mainly concerned with investigating the conditions and estimating the effects of these components of transfer.

4.2 Components of Learning

On the side of *specific* transfer effects, namely the effects of associative interference, five components were distinguished on theoretical grounds (see Postman, 1971, p. 1060). First, if the responses are meaningless letter sequences they have to be acquired independently from their association with a particular stimulus. This is what has been termed response learning (see Table 5 (a), Underwood & Schultz, 1960). Furthermore, Postman lists stimulus and response discrimination (b,c) as possible sources for transfer. He argues that in a list learning experiment, responses as well as stimuli have to be discriminated from each other, before they can be successfully associated to their respective associates. List discrimination acquired in first list learning can therefore transfer to learning to the second list, because the members of one list have already been discriminated from each other. Finally, Postman distinguishes between two truly associative components of transfer: (e) forward and (f) backward associations. These are the learned associations between two memory items that have been established during the learning of the first list. Forward associations are expected to interfere with learning of new responses to old stimuli, because the stimulus evokes the old response association during learning of the second list. Forward associations may have a positive effect when the new responses are similar to the old responses. Backward associations are the effect by which a response (B) learned in context with a particular stimulus (A-B) will evoke the stimulus (A). This might have an interfering effect whenever an old set of responses has to be associated with a new set of stimuli (C-B). Postman describes this effect as a possible withholding of the correct response when the first list associations are elicited during second list learning, where the backward associations serve a 'checking function' to edit responses (Twedt and Underwood, 1959, McGovern, 1964).

These 'specific' transfer components were distinguished from two effects called 'unspecific' or *general* transfer. Concern for the effect of these transfer components came from unpredicted empirical results. For example, transfer was observed in the basic control condition (A-B, C-D). If transfer is conceptualized as the effect of interfering memory elements, transfer should not be expected when learned list items are not identical or even similar to each other (as in A-B, C-D). Therefore these transfer effects were puzzling. Similarly, in experiments where negative transfer had been predicted as an effect of associative interference, (A-B, A-D), some researchers reported a failure to establish the predicted effects (Heron, 1928). However, Heron observed that all second lists in his experiment were learned faster, when they were acquired during the same session. Subjects did not show any

improvement from one session to the next, though. Such findings led to experimental investigations of the general transfer component.

		Components of Transfer effects in basic designs				
		C-D	A-B'	A-D	C-B	A-Br
<u>Unspecific components</u>						
	(a) warm up	+	+	+	+	+
	(b) learning to learn	+	+	+	+	+
<u>Specific components</u>						
	(a) response learning	0	+	0	+	+
	(b) response differentiation	0	+	0	+	+
	(c) stimulus differentiation	0	+	+	0	+
	(e) forward associations	0	+	-	0	-
	(f) backward associations	0	+	0	-	-

Table 5. Components of Transfer

Again one might simply reliable the unspecific transfer components with 'procedural transfer', because the experiments that describe learning to learn, suggest that it is very similar to characteristics of procedural skill acquisition. The specific components that relate interference and transfer performance to particular variations in the experimental material may be re-labeled 'declarative component'. In fact, interference has been explained by the fan effect, related to certain conditions of declarative learning and retrieval (see chapter 2.)

We will now describe characteristics of each of the components in some detail.

4.2.1 Learning to Learn

Learning to learn is defined as the acquisition of learning skills and habits that are specific for a particular class of tasks, such as paired associate learning or serial list learning. An example is transfer in the A-B, C-D paradigm (as discussed above), where training on a previous paired-associate task leads to better performance on succeeding paired associate learning even though no transfer would be predicted on grounds of the specific transfer components. Learning to learn was shown to have certain characteristic effects, that were investigated in detail.

4.2.1.1 Learning to learn as a negatively accelerated function of amount of practice.

Postman reports several independent experiments that show that improvement on a particular task is essentially a negatively accelerated function of the amount of practice on that task. Improvement is strong during the first trials on a task, and levels off in later trials. Evidence for this comes from experiments involving serial learning of nonsense syllables (Ward, 1937), and words and numbers (Melton & von Lackum, 194, and Postman, 1962), as well as paired associate learning of adjectives (Greenberg &

Underwood, 1950), and nonverbal learning involving visual patterns as stimuli and lever presses as responses (Duncan, 1950). Task difficulty (in terms of variation in the material or higher performance criteria) has influence on the number of trials until asymptote is reached, but no effect on the general shape of the practice function (Melton & Von Lackum, 1941).

4.2.1.2 Effects of varied and constant practice. If continued practice on a particular tasks is given, transfer to a test task will be greater when the subjects were trained with varying sets of stimuli, as when they were trained with the same set of stimuli. Evidence for this effect comes from an experiment by Duncan (1958), in which paired associates between visual patterns and lever presses had to be learned. Practice effects were found to be greater when subjects were trained on varying stimulus sets during practice. When only practice is not varied however (as in an experiment by Postman & Schwartz, 1964), transfer is more extreme when the training task involved the same learning method (serial learning or paired-associate learning), or the same learning material (adjectives or trigrams), as the transfer task. However, lower levels of transfer are even found for practice on the different method or material as compared to a no practice control condition. (Postman & Stewards, 1964).

4.2.1.3 Organizational skill in free recall learning. Finally, several free recall experiments show that improvement in the ability to recall a list of items can be related to an increasing use of subjective organization in the recall of the items. Evidence for this effect is reported in studies by Dallett (1963), Tulving, McNulty & Ozier (1965), and Mayhew (1967). In Mayhew's study, subject learned two lists of 30 words per day on three consecutive days. One group of subjects were informed about useful ways to organize the items and encouraged to use them, whereas the other group of subjects did the experiment naive. For both conditions the recall probabilities increased at the same rate as subjective organization scores, but the learning curve was more accelerated for the informed group.

4.2.1.4 Learning to learn under modern perspective. These three characteristics of learning to learn suggest that this component is indeed the same that has been studied in the modern literature under the label skill acquisition. First, the negatively accelerated practice function is essentially the Power Law of Practice, that has been found to describe the acquisition of any cognitive or motor-perceptual task (see for example Card, Moran & Newell, 1983). It states, that speedup on a particular task is approximately proportional the power of the amount of practice given on that task. The function has been repeatedly used to describe the acquisition of various cognitive skills, such as editor use or learning to code in lisp (i.e. Polson, 1988, and Singley & Anderson, 1989). The mechanism in Soar that explains how this architecture would learn paired-associates in recall-problem spaces is a good hypothesis about the type of procedural skill that may be learned to facilitate paired-associated chunking.

Second, the effects of varied and continuous practice prior to transfer has been examined in the modern literature under the headers 'schema-building' or 'generalization of rules'. The general argument goes like this: If continuous training on a particular task is given, training with varying material will produce more general productions, that in turn will be applicable to a broader set of new material. If only training on one prior task is provided, the newly learned productions are specific to their use in that previous context. Therefore, transfer on the next trial will be greatest, when the material is exactly the same. If the material is slightly different from the previous, then the old production would have to be modified, which could be expected to take some additional learning time (see Polson, et al., 1985). In the modern work it has been difficult to find evidence for the effect of generalization on transfer performance (i.e. Polson, 1988). The data reported here then provide some 'new' evidence for the existence of this effect during learning. We will address accounts of this learning process in a later chapter.

Third, moderate transfer between two slightly different methods (serial vs. paired associate learning) is most likely due to the application of a subset of productions that have been acquired in the learning task. This interpretation is supported by the observation that training on "paired-associate learning disposed the subjects to subdivide the series (of words) into a sequence of pairs." (Postman, 1971, p. 1043. This is clear evidence for transfer of a production that was learned in one task (paired associate learning) to a similar task (serial list learning).

Finally, increased performance in free recall produced by increased 'subjective organization', has been described in detail as the build up and use of elaborate memory structures, that serve in encoding and retrieval of presented information (see Chase & Ericsson, 1985).

4.3 Specific Transfer components

The specific components of transfer as identified by Postman can be roughly divided into two groups by their emerging during different learning phases (Underwood & Schultz, 1960). First, the new stimuli and responses have to be learned as independent events, and items within lists have to be distinguished. This phase of learning entails response learning, stimulus and response differentiation. After stimuli and responses have become available as independent events, associations between them can be established. This is the associative phase of learning in which forward and backward associations are acquired. In the following sections evidence for the effect of the proposed specific transfer components (see Table 5) will be reviewed briefly. In a final section it will be considered whether and how these transfer components might be incorporated into a analysis based on a production system analysis.

4.3.1 Response Learning. The literature provides rich evidence for the importance of response learning. Experiments demonstrating the effect rely

on the argument that subjects need longer time to learn low meaning responses than high meaning responses. (for example trigrams instead of words). Therefore the positive transfer effect of the response learning component should be stronger when low meaning responses are transferred to the second task (C-B, A-Br, A-B'), as when high meaning responses are used. This result for the C-B and A-B' paradigm is reported in the literature (Jung, 1963; and Merikle, 1968). Jung compared the transfer of low and high meaning trigram responses in the A-B, C-B paradigm. Transfer was positive for low, and slightly negative for high meaning responses. Merikle produced the same result with the same type of material using a A-B, A-B' paradigm.

4.3.2 Stimulus differentiation. Postman (1971) reports only weak evidence for the effect of stimulus differentiation. In a study by Underwood and Ekstrand (1968) subjects were trained in an A-B, A-D paradigm with varying intralist similarity of A. The experiment relies on the assumption that stimulus differentiation is more difficult with items that are highly similar to each other. Therefore the transfer effect achieved for the high similarity stimuli should be more positive than for the stimulus lists that are clearly distinguishable. Underwood and Ekstrand's data show this predicted trend, but the effects lack statistical significance.

4.3.3 Response differentiation. Postman (1971) does not report any data that bear directly on the existence of this transfer component. However, the appropriate design to study the phenomenon should combine the experimental logic of the two paradigms reported above. In a factorial design, the two factors meaningfulness of responses and intralist similarity of responses should be combined. The appropriate transfer paradigm would be A-B, C-B. If response differentiation has an independent effect from response learning, then there should be a main effect for intralist similarity and meaningfulness, but no interaction between the two factors.

4.3.4 Forward Associations. The data reported by Postman show only weak support for the effect of the associative component on transfer. The design that would isolate this component most effectively is A-B, A-D with meaningful stimulus lists that have low intralist similarity. The interfering effect of forward associations should increase with higher levels of first list learning in this design, where the effect of stimulus differentiation is kept at a minimum. An experiment by Mandler and Heinemann (1956) included one condition similar to this, where the stimuli had low meaning. The data show a trend toward higher levels of interference after longer first list learning, however this trend is not significant.

4.3.5 Backward Associations. Evidence for the effect of backward associations comes from the same experiment (Mandler & Heinemann, 1956), and again does not yield statistical significance. The A-B, C-B paradigm would isolate negative interference based on backward associations best. Again, the data show a trend towards more negative transfer at high levels of overlearning. However, the same criticism of the design applies as for the

condition representing the A-B, A-C design: Highly meaningful and easily distinguishable responses would have minimized the positive effect of response learning and differentiation, however low meaning items were used in the experiment.

4.3.6 Summary The review of this literature shows that it has been difficult to provide clean evidence for the importance for each of the proposed specific components, even in this paradigm, where it is relatively easy to control the important characteristics of the involved material. However, many researchers have agreed (i.e. Kintsch, 1963; Polson, Restle & Polson, 1965; Greeno, James, & DaPolito, 1971, and Martin, 1971) that paired-associate learning can be described by two phases, where the first phase is attributed to some type of learning or encoding of the independent events, and the second phase is attributed to learning appropriate associations between them, or in other words establishing efficient retrieval structures.

As in the examples from the procedural learning literature, the first, stimulus learning phase is of particular importance when the involved material is low in initial meaningfulness (such as lists of Lisp expressions, or a new set of logic rules), and when the set of new concepts or items is difficult to differentiate. In most situations involving procedural skill in a realistic domain, both of these characteristics will be confounded. Whenever a new set of terms is introduced it will be difficult for the learner to distinguish between them. With the acquisition of a semantic representation, each item in the set will be easier to recognize, as well as it will be easier to distinguish from other items in the set.

We have shown that two general components, one that could be termed procedural and another that can be called declarative combine to explain transfer in the domain of verbal learning. This result combined with the evidence from the current transfer work emphasizes that transfer has to be analyzed under consideration of both of these factors.

Furthermore, there is evidence from both research directions, that the declarative learning phase might consist of two components or sub processes. One of these components is connected to establishing some representation of the new concepts, whereas the second phase is responsible for learning particular associations (of either conditions and actions, or of particular concepts from this domain).

Chapter 5. General Conclusions

In the present paper we discussed learning and transfer in the procedural domain, as well as learning and transfer of simple declarative material such as paired-associates. Procedural learning has been characterized as the transition of two main learning phases: a problem solving phase in which appropriate condition actions pairs are discovered in a declarative representation of the problem space. Once these appropriate pairings have been established ('compilation' in Act*), further learning is a process of tuning and adaptation of these production rules for special cases of application and a combination of smaller rules into larger units for frequently repeated sequences of steps.

5.1 Procedural vs. declarative learning

A previously presented argument (Anderson, 1983, 1987; and Polson, 1988) was that transfer of procedural skill is the sole effect of whole-cloth transfer of such compiled rules, where general rules may be applied in a range of similar problems, and specific rules apply only to very similar situations. The review of recent transfer experiments however shows that initial problem solving time may vary greatly dependent of the state of the declarative representation of a given domain. If a domain is already well represented (such as the key concepts in the text editing domain: moving, printing, deleting), general problem solving productions may access relevant parts of this representation and derive correct solution steps in a relatively efficient way. However, if the concepts are not accessible, or if no problem relevant representation of the new concepts is given, these problem solving mechanisms may have nothing to work with. It may be that in such a case much more elementary problem solving, or concept learning mechanisms have to be invoked before a meaningful representation of the concepts can be established.

We argued that in such cases (such as in the experiments in the domains of Lisp programming, and Calculus and Logic reviewed in Chapter 3), transfer subjects can benefit from an improved representation of the new domain, even if there are few shared productions between the subtasks, such as between generating and evaluating Lisp code. If part of what was acquired during original learning was to establish a rich declarative representation of the new domain, then transfer subjects can benefit from this newly acquired declarative knowledge representation.

5.2 Outlook

We pointed out that further research needs to address the distinction between transfer based on common productions and common declarative knowledge more clearly. In most of the current research, the newly acquired

declarative knowledge may overlap with the productions that have been proposed to account for the transfer effect. Properly designed experiments should address whether the transfer results were in fact due to applicable productions or whether declarative transfer can also be held responsible for these results.

Furthermore, the inter-dependencies between problem solving and acquisition of declarative knowledge need to be understood more clearly. This point can be decomposed into two important questions. First, we need to understand what characteristics of a declarative representation are most important for skill acquisition in a new domain. Is it important that new concepts become well elaborated, and is it important that this elaboration is problem-relevant? A rich elaboration of key concept may influence their retrievability negatively (via fan-effect). Is accessibility of new concepts more or less important than the semantic richness of their representation? Second, what are the conditions under which a new conceptual structure can be acquired? Should concept learning be understood as problem solving? Which types of tasks lead to a fast and effective acquisition of concepts that furthermore aid procedural learning optimally? In other words, a finer grained analysis of the interaction between concept learning and problem solving may be indicated.

References.

- Anderson, J.R., 1976. Language, memory, and thought. Hillsdale, N.J.: Erlbaum Associates.
- Anderson, J.R., 1983. The architecture of cognition. Cambridge, Mass.: Harvard University Press.
- Anderson, J.R., 1986. Acquisition of cognitive skill. *Psychological Review*, 89, 369-406.
- Anderson, J.R., 1986. Knowledge compilation: the general learning mechanism. In R. Michalski, J. Carbonell, and T. Mitchell, (eds.), *Machine learning*, Vol. 2. pp. 289-310. Los Altos, Calif.: Morgan Kaufmann.
- Anderson, J.R., 1987. Skill acquisition: compilation of weak-method problem solutions. *Psychological Review*, 94, 192-210.
- Angell, J.R., 1908. The doctrine of formal discipline in the light of the principles of general psychology. *Educational Review*, 36, 1-14.
- Bovair, S., Kieras, D.E, and Polson,P.G., 1990. The acquisition and performance of text-editing skill: a cognitive complexity analysis. *Human Computer Interaction*, 5, 1-48.
- Card, S.K., Moran,T.P., and Newell ,A., 1983. The psychology of human-computer-interaction. Hillsdale, N.J.: Erlbaum.
- Chase, W.G., and Ericsson, K.A., 1982. Skill and working memory. In G.H. Bower (Ed.), *The psychology of learning and motivation*, Vol. 16, pp. 1-58, New York, Academic Press.
- Conrad, F., and Anderson, J.R., 1988. The process of learning Lisp. In *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*, Montreal.
- Dallett, K.M., 1963. Practice effects in free and ordered recall. *Journal of Experimental Psychology*, 66, 65-71.
- Duncan, C.P., 1958. Transfer after training with single versus multiple tasks. *Journal of Exprimenal Psychology*, 55, 63-72.
- Engelbeck, G. 1989. An integrated model of the initial acquisition and retention of a cognitive skill. Unpublished doctoral thesis. (Tech. Rep. #89-2). Boulder: University of Colorado, Institute of Cognitive Science.

- Foltz, P.W., Davies, S.E., and Polson, P.G., 1988. Transfer between menu systems. *Proceedings CHI '88 Human Factors in Computing Systems*, (pp.107-112).
- Greenberg, R., and Underwood, B.J., 1950. Retention as a function of stage of practice. *Journal of Experimental Psychology*, 40, 452-457.
- Greeno, J.G., James, C.T., and DaPolito, F.J., 1971. A cognitive interpretation of negative transfer and forgetting of paired-associates. *Journal of Verbal Learning and Verbal Behavior*, 10, 331-345.
- Hamilton, C.E., 1950, The relationship between the length of interval separating two learning tasks and performance on the second task. *Journal of Experimental Psychology*, 40, 613-621.
- Heron, W.T., 1928. The warming-up effect in learning nonsense syllables. *Journal of Genetic Psychology*, 1928, 35, 219-228.
- Irion, A.L., 1949. The relation of "set" to retention. *Psychological Review*, 55, 336-341.
- Jung, J. 1963. Effect of response meaningfulness (m) on transfer of training under two different paradigms. *Journal of Experimental Psychology*, 65, 377-384.
- Kessler, C. 1988. Transfer of programming skills in novice Lisp learners. Unpublished Doctoral dissertation, Carnegie-Mellon- University.
- Kessler, C., and Anderson, J.R., 1988. Learning flow of control: recursive and iterative procedures. *Human -Computer Interaction*, 2, 135-166.
- Kieras, D.E., and Polson, P.G., 1985. An approach to the formal analysis of user complexity. *International Journal of Man-Machine Studies*, 22, 365-394.
- Kintsch, W., 1963. All-or-none learning and the role of repetition in paired associate learning. *Science*, 140, 310-312.
- Laird, J.E., Rosenbloom, P.S., Newell, A., 1986. Universal subgoalting and chunking: the automatic generation and learning of goal hierarchies. Boston; Kluwer.
- Lee, A.Y., Polson, P.G., and Bailey, W.A., 1989. *Proceedings CHI '89 Human Factors in Computing Systems*, (pp.115-120).
- Lewis, M.W., and Anderson, J.R., (Forthcoming). Declarative and procedural transfer in logic. Reported in Singley, M.K., Anderson, J.R., 1989. The

transfer of cognitive skill, (chapter 7). Cambridge, Mass.: Harvard University Press.

- Lewis, R.L, Newell, A., and Polk, T.A., 1989. Toward a Soar theory of taking instructions for immediate reasoning tasks. Proceedings, Cognitive Science Annual Conference. University of Michigan, Ann Arbor. Cognitive Science Society.
- Mandler, G., and Heinemann, S.H., 1956. Effect of overlearning of a verbal response on transfer of training. *Journal of Experimental Psychology*, 57, 39-46.
- Martin, E., 1965. Transfer of verbal paired associates. *Psychological Review*, 72, 327-343.
- Martin, E., 1971. Verbal learning theory and independent retrieval phenomena. *Psychological Review*, 27, 314-332.
- Mayhew, A.J., 1967. Interlist changes in subjective organization during free-recall learning. *Journal of Experimental Psychology*, 74, 425-430.
- McGovern, J.B., 1964. Extinction of associations in four transfer paradigms. *Psychological Monographs*, 78.
- McKendree, J.E., and Anderson, J.R., 1988. Frequency and practice effect on the composition of knowledge in Lisp evaluation. In J.M. Carroll (ed.), *Cognitive aspects of human-computer interaction*. Cambridge, Mass.: MIT Press.
- Melton, A.W., and von Lackum, W.J., 1941. Retroactive and proactive inhibition in retention: Evidence for a two-factor theory of retroactive inhibition. *American Journal of Psychology*, 54, 157-173.
- Merikle, P.M., 1968. Paired-associate transfer and response meaningfulness. *Psychological Reports*, 22, 131-138.
- Müller, G.E., and Schumann, F., 1984. Experimentelle Beiträge zur Untersuchung des Gedächtnisses. *Zeitschrift für Psychologie*, 6, 81-190.
- Newell, A., 1980. Reasoning, problem solving, and decision processes: The problem space as fundamental category. In R. Nickerson (ed.), *Attention and Performance VIII*. Hillsdale, N.J. : Erlbaum.
- Newton, J.M., and Wickens, D.D., 1956. Retroactive inhibition as a function of the temporal position of interpolated learning. *Journal of Experimental Psychology*, 51, 149-154.

- Osgood, C.E., 1949. The similarity paradox in human learning: a resolution. *Psychological Review*, 56, 132-143.
- Pennington, N., and Nicolich, R., (in press). Transfer of training between programming subtasks: is transfer really use specific? *Empirical Studies of Programmers*, Vol. 4.,
- Pennington, N., Nicholich, R., and Rahm, J., (submitted). Transfer of Training between cognitive subskills: is knowledge use specific?
- Polson, P.G., 1988. The consequences of consistent and inconsistent interfaces. In R. Guindon (Ed.), *Cognitive science and its application for human computer interaction*. Hilldale, NJ: Lawrence Erlbaum Associates.
- Polson, P.G., Bovair, S., and Kieras, D.E., 1987. Transfer between text editors. In P. Tanner & J.M. Carroll (Eds.), *Proceeding CHI'87 Human Factors in Computing Systems* (pp. 27-32). New York: ACM.
- Polson, P.G., Muncher, E., and Engelbeck, G. 1986. A test of a common elements theory of transfer. In M. Mantei & P. Orbeton (Eds.), *Proceedings CHI'86 Human Factos in Computer Systems* (pp. 78-93). San Francisco: ACM.
- Polson, P.G., Muncher, E., and Kieras, D.E., 1987. Transfer of skills between inconsistent editors. (Tech. Rep. # 87-10). Boulder: University of Colorado, Institute of Cognitive Science.
- Polson, M.C., Restle, F., and Polson, P.G., 1965. Association and discrimination in paired-associate learning. *Journal of Expeimental Psychology*, 69, 47-55.
- Postman, L. (1971). Transfer, interference, and forgetting. In J.W. King and L.A. Riggs (eds.), *Woodworth and Schlosberg's experimental psychology* (pp. 1019-1132). New York: Holdt, Rinehart, and Winston.
- Postman, L., 1962. The temporal course of proactive inhibition for serial lists. *Journal of Experimenal Psychology*, 1962, 63, 361-369.
- Postman, L., and Schwartz, M., 1964. Studies of learning to learn: 1. Transfer as a function of method of practice and class of verbal materials. *Journal of Verbal Learning and Verbal Behavior*, 3, 37-49.
- Rosenbloom, P.S., Laird, J.E., Newell, A., McCarl, R., 1991. A preliminary analysis of the Soar architecture as a basis for general intelligence. *Artificial Intelligence*, 47, 289-325.

- Schwenn, E., and Postman, L., 1967. Studies of learning to learn: V. Gains in performance as a function of warm-up and associative practice. *Journal of Verbal Learning and Verbal Behavior*, 1967, 6, 565-573.
- Singley, M.K., and Anderson, J.R., 1985. The transfer of text-editing skill. *Journal of Man-Machine Studies*, 22, 403-423.
- Singley, M.K., and Anderson, J.R., 1988. A keystroke analysis of learning and transfer in text editing. *Human-Computer Interaction*, 3, 223-274.
- Singley, M.K., Anderson, J.R., 1989. The transfer of cognitive skill. Cambridge, Mass.: Harvard University Press.
- Steier, D.M., Laird, J.E., Newell, A., Rosenbloom, P.S., Flynn, R.S., Golding, A., Polsk, T.A., Shivers, O.G., Unruh, A., Yost, G.R., 1987. In P. Langley (ed.), *Proceeding Forth International Workshop in Machine Learning*, Irvine, CA, pp. 300-311.
- Thorndike, E.L., and Woodworth, R.S., 1901. The influence of improvement in one mental function upon the efficiency of other functions. *Psychological Review*, 9, 374-382.
- Thune, E.L., 1950. The effect of different types of preliminary activities on subsequent learning of paired-associate material. *Journal of Experimental Psychology*, 40, 423-438.
- Thune, E.L., 1951. Warm-up effect as a function of level of practice in verbal learning. *Journal of Experimental Psychology*, 42, 250-256.
- Tulvin, E., McNulty, J.A., and Ozier, M., 1965. Vividness of words and learning-to learn in free recall-learning. *Canadian Journal of Psychology*, 19, 242-252.
- Twedt, H.M., and Underwood, B.J., 1959. Mixed vs. unmixed lists in transfer studies. *Journal of Experimental Psychology*, 48, 111-116.
- Underwood, B.J., and Ekstrand, B.R., 1968. Differentiation among stimuli as a factor in transfer performance. *Journal of Verbal Learning and Verbal Behavior*, 7, 172-175.
- Underwood, B.J., and Schultz, R.W., 1960. *Meaningfulness and verbal learning*. Chicago: Lippincott.
- Ward, L.B., 1937. Reminiscence and rote learning. *Psychological Monographs*, 49, No. 220.

- Wharton, C., and Lewis, C. 1990. Soar and the construction-integration model: Pressing a button in two cognitive architectures. Technical Report # CU-CS-466-90. University of Colorado.
- Ziegler, Hoppe, and Fähnrich, 1986. Learning and transfer for text and graphics editing with a direct manipulation interface. In M. Mantei & P. Orbeton (Eds.), Proceedings CHI'86 Human Factors in Computer Systems (pp. 72-77). San Francisco: ACM.

