Simulating the interaction of explicit and implicit learning using an integrated cognitive model

Masters thesis presented in partial fulfilment of degree requirements

Pawel Slusarz
Thesis advisor: Ron Sun, Ph.D.

University of Missouri - Columbia
Computer Engineering - Computer Science Department

May 16, 2003

Abstract

Since the 1980s, experimental psychologists, using a number of cognitive and motor skill learning tasks, demonstrated a contrast between improved task performance and the lack of corresponding explicit knowledge in human learning. Experiments have confirmed that in the situation when no apriori knowledge about the nature of the task is available, human subjects implicitly learn to improve performance and only develop explicit knowledge afterwards. One explanation for this phenomenon could be the existence of two separate structures in the human mind: one implicit, distributed, and not consciously available, and another explicit, crisp, and available for conscious retrieval. In the field of cognitive modeling, computer-assisted models are developed to simulate human learning, test hypotheses, and gain insight into cognitive processes. CLARION is a model based on the assumption of , which uses a Q-learning neural network to represent the implicit learning structures, and a rule system to represent the mind's explicit part. This work presents CLARION simulations of a number of human process control task experiments, and examines the model's viability for simulating human learning when no apriori knowledge is available.

Contents

Acknowledgements

Dr. Ron Sun, of the University of Missouri-Columbia has been the thesis advisor for this thesis paper. He is the designer of CLARION and his research has been instrumental in the development of this work.

Experiments simulated here were presented in a conference paper. The paper is:

> P. Slusarz and R. Sun, (2001). The interaction of explicit and implicit learning: Exploring an integrated computational model. *Proceedings of the 23rd Cognitive Science Society Conference, Edinburgh, 2001.* pp.952-957. Lawrence Erlbaum Associates, Mahwah, NJ.

Another paper in which this research was included is:

> R. Sun, P. Slusarz, and C. Terry, (2002). The interaction between the implicit and explicit processes: A dual process approach. Submitted for publication.

Chapter 1

Motivation and background

1.1 Motivation

Biological organisms have been an inspiration for the study of machine learning, leading to development of such techniques as artificial neural networks, genetic algorithms, and reinforcement learning. But those same machine learning algorithms can in turn also be used to aid in understanding of how biological organisms function. For the obvious reasons, one area of research interest is human learning, with its immediate implications in the area of education, but also the ultimate goal of understanding the nature of the mind. By constructing computer models which employ machine learning algorithms to simulate human learning, scientists can verify the performance of such algorithms against their biological equivalents, and at the same time gain important insight into the nature of the processes being simulated.

This work will examine in greater detail one type of learning, prevalent in situations where there is little apriori knowledge about the task, in skill learning, and in situations where attention is occupied by other matters. The remainder of this chapter will be devoted to the introduction of the dual approach to learning, and the presentation of the dual representation hypothesis, a framework which can be used to study the phenomenon. This framework forms the basis of a computational cognitive model, which is presented in the next chapter. The model is then used to simulate a number of human experiments, to which chapters 3 and 4 are devoted. The model optimization and sensitivity to parameter manipulation is then studied in greater detail in chapter 5, and finally conclusions and closing remarks are presented.

1.2 Dual approach to learning

Parents often urge their children to attend school and then continue their education in an institution of higher learning. Students in such settings are presented with knowledge in the form of lectures and textbooks, and then they are tested over how much of this knowledge they have retained. The ultimate goal of this learning process is to provide enough theoretical knowledge to the students so that they could become experts in certain areas, and perform well in "real life" tasks.

However, theoretical knowledge acquisition through verbal or written instructions, is only one way in which humans can learn. In addition to the school learning model, humans can gain knowledge either without being consciously aware of it, or unable to explicitly communicate it to others, in a process which only becomes apparent during performance tests. Such type of learning is particularly apparent during motor skill acquisition. No matter the amount of theory given beforehand, no novice will be able to ride a bicycle on their first try. However, after a number of trials, novices will aquire the sufficient degree of balance, steering coordination, and speed control in order to ride a bike competently. Such expert riders are then able to convey certain general principles of bike riding to novices, but those novices will not be able to ride the bike on their first tries.

1.2.1 Implicit and explicit learning

The significant skill improvement which takes place from the time a novice is given verbal knowledge from a competent bike rider to the time he is able to ride the bike competently himself, is when what is known as implicit learning dominates the learning process. In contrast, if a teacher were to give a presentation on how to perform long division, and then started students on doing example problems by themselves, learning would have been explicit.

Explicit and implicit learning are not separated along the motor-mental skill axis. When learning to throw a ball, one may discover that in order for the baseball to travel furthest, one needs to throw it at about 45 degree angle with respect to the ground. This would be an explicit rule, which would siginificantly improve performance, and could be conveyed to other would-be ball throwers. On the other hand, a History teacher may notice that some his students have developed a more mature writing style throughout the process of completing a series of essay assignments on historic subjects. Confronting those students, the teacher may then find out that the students would be unable to explicitly state the stylistic differences between their earlier and latter essays, thus demonstrating that implicit learning may have taken place thoughout the process of essay writing.

Implicit and explicit learning are best defined by the type of knowledge that is acquired during the process. When the knowledge can be verbalized and conveyed to others, we speak of explicit learning. When knowledge cannot be conveyed in words, but nevertheless performance improves, implicit learning is taking place.

1.2.2 Procedural and declarative knowledge

Another related perspective on learning is that of declarative and procedural knowledge. While learning to throw the ball, one might be told to aim at a 45 degree angle in order to get it to travel the furthest. By practicing, the would-be olympic ball thrower would learn to lean back slightly, align his arms, and then swing and release the ball at the most convenient moment. In this case, first came the acquisition of explicit knowledge concerning the task, and the knowledge was of general nature. This kind of information is known as declarative knowledge. In order to perform the task it was necessary to process declarative knowledge and consciously apply it. With time and practice, a set of practical procedures was developed which made it unnecessary to consult declarative knowledge and throw the ball with little conscious intervention. Those automated procedures are refered to as procedural knowledge. The above paradigm can also be discussed on the basis of the long division example, and it can be seen how it is widely present in the math classroom environment.

The exact nature of a relationship between explicit and implicit knowledge on one side, and declarative and procedural knowedge on the other has not been established in literature. One important distinction is that procedural knowledge may be consciously available, while its common points with implicit knowledge are that both can be applied with little conscious intervention, and both are practically oriented. On the other side of the spectrum, while all declarative knowledge is explicit, some explicit knowledge could be practically oriented, and therefore procedural in nature.

1.2.3 Bottom-up and top-down learning

In practice, implicit and explicit learning occur together on every task. When asked to explain how to ride a bike, experienced bikers are eventually able to provide useful advice to novices, which indicates that some explicit learning has taken place, even if the implicit learning has dominated the bike riding skill acquisition. Similarly, a student who has forgotten lecture material, will still perform better on a test than a student who has never been to the lecture.

Nevertheless, there is a substantive difference between acquiring knowledge on solving a particular math problem, and then becoming proficient at it, and learning a new motor skill and then giving tips to someone else. In the former case, declarative knowledge is acquired first, and then procedural knowledge is formed as a compliment during practice. In the latter case, procedural knowledge is formed first, and then some of it is eventually formalized as declarative knowledge.

The type of learning which takes place when implicit learning is primarily responsible for performance improvement, is called bottom-up learning. During the bottom-up learning procedural knowledge is "pumped up" to the conscious level, leading to declarative knowledge, which gives this type of learning its name. In contrast, during top-down learning, declarative knowledge is used as a basis for the formation of procedural knowledge. The knowledge "leaks down" from conscious to the level not available for verbalization. Top-down learning is dominated by explicit processes, and implicit learning is only used as a supplement. This work is only concerned with the relatively little researched bottom-up learning, the kind most known for its presence during motor skill acquisition.

Of course it is often difficult to determine which kind of knowledge was formulated first or dominant in learning given skill. A bike rider can receive detailed instructions on riding a unicycle, and based on these instructions and hie procedural knowledge of bike riding, pick up unicycle riding with little effort. In other cases declarative, "common sense" knowledge can be present prior to implicit learning taking place. Such is the probable explanation for better subject performance on personal interaction task than factory production taks during experiments presented in Stanley et al (see 3) [26]. Finally, different types of learning can be dominant during different stages of skill acquisition, and the knowledge can pass both"up" and "down" during the learning process. When researching a given skill acquisition (or a particular stage thereof) we must accept the fact the learning taking place will lie on a continuum somewhere between top-down and bottom-up, its position at a given time depending on the dominance of explicit or implicit processes.

1.3 Studies of the two types of learning

Philosophical origins of the dichotomy of human cognition can be traced to Heidegger, and his ideas on ontological and preontological [9], and even further back to William James and his distinction between "empirical thinking" and "true reasoning" [11]. However, it was not until much later that the ideas have been investigated in Cognitive Science.

Implicit and explicit learning has been studied by Reber et al. in his work on artificial grammar learning (for more details, see section 4.1 on page 30), some 100 years after James. In a classic experiment, participants are presented with a series of strings generated by a finite state automaton, and therefore following artificial grammar rules [19]. After having studied the strings, subjects were told that the strings were generated according to certain rules, and then they were presented with a series of random strings, as well as ones conforming to the grammar (but different than those in the first phase), and asked to identify strings which violated the grammar. Human subjects had suprisingly good results in this task. When studying the initial strings they were not aware of any rules governing their structure, and, since none of the initially studied strings was given in the testing phase, Reber concluded that the grammar was learned implicitly.

In 1970, Warrington and Weiskrantz studied the learning of patients suffering from amnesia [33]. They found that on explicit tasks like recall and recognition, amnesia patients perform much worse than healthy individuals. On the other hand, on implicit tasks, like word completion, performance of amnesia patients was as good as that of healthy people.

In another interesting study, conducted in 1982, Tulving et. al. sought to create a dissociation between the two types of learning [32]. In the experiment, the subjects were first asked to study a list of 96 words. They were then fist tested by being asked to identify whether a word belonged to the original list of 96 study words or not. Then a second testing method was used: given a word fragment with some letters filled in, subjects were asked to identify the word by filling in the missing letters. Both tests were repeated a week later. Immediately following the experiment the results of the first, more explicit test and the second test were significantly above chance. However, a week later performance on the explicit test dropped dramatically, while performance on the implicit test remained at the same level.

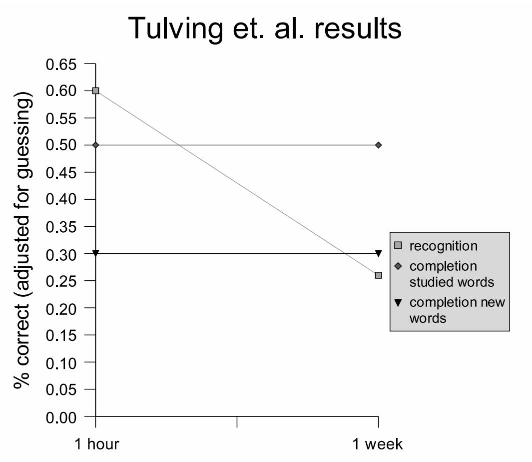

Figure 1.1: Results of word learning experiment by Tulving et. al.

In terms of accessibility, in 1988, Smolensky has distinguished between publicly accessible and inaccessible processing of information [24]. He called those conceptual and subconceptual modes of processing, by analogy to conscious and subconscious, which fits in with Reber's ideas.

In 1983, Anderson has coined the terms procedural and declarative knowledge [1]. He investigated skill learning, where the initial stage was acquisition of verbal knowledge in performing the task. At first, it'd be necessary to consciously access this knowledge in order to perform the task, but with time, a sort of automatic processing would develop, where the task could be performed without accessing the declarative knowledge. Anderson refered to this second type of knowledge as procedural, since it involved developing procedures and then following the routine to perform the task. This study is an example of what is refered to here as top-down learning. [1] Another such

[1]In [27] Sun writes:

> The reason for having both action-centered and non-action-centered modules (...) is that (...) action-centered knowledge (roughly, procedural knowledge) is not necessarily inaccessible directly, and non-action-centered knowledge (roughly, declarative knowledge) is not necessarily accessible directly. Although it was argued by some (...) that all procedural knowledge

example is discussed in Dreyfus and Dreyfus [7]. In their analyses of chess player skill learning, they found that the beginner chess players would try to apply explicit rules into decision making, while advanced players relied more on intuition in selecting the right moves. To describe this process, they proposed the distinction between analytical and intuitive thinking [7]. [2]

1.4 Theories explaining the implicit-explicit distinction

There are several approaches to explaining the distinction between explicit and implicit learning apparent in experimental findings. One approach is to provide explanation in terms of a single underlying structure. For example, Berry and Broadbent explained that learning could either take into account all possible variables, and progress slower, or concentrate only on a few select variables, allowing for either fast progress, or no progress at all if irrelevant variables were chosen [6]. In another work, Lebiere et al. explain the implicit-explicit distinction through the existing ACT-R cognitive model architecture [3] [14].

Rather than come up with complicated explanations for the experimentally measurable differences between implicit and explicit learning, another approach is to allow the possibility that the two types of learning are manifestations of two different structures in the mind responsible for learning. The processing theory distinguishes between data and conceptually driven processeses in the mind [21]. Conceptual processes are initiated by the mind and lead to explicit learning, while data driven processes are externally stim-

is inaccessible directly and all declarative knowledge is directly accessible, such a clean mapping of the two dichotomies is untentable in my view.

While the two are related, there remains a distinction between implicit-explicit processes and procedural-declarative ones, and the refering passage should not be construed to indicate otherwise. Anderson's study was mentioned here, because while it focuses on procedural and declarative knowledge, it also provides information about implicit and explicit learning.

[2] It should be noted that for a long time cognitive research on learning has been dominated by such top-down approaches (e.g., [22], [3], [4]). The bottom-up direction, or learning both in parallel has been largely ignored. There are a few notable exceptions, which investigated parallel development of the two types of knowledge or the extraction of declarative knowledge from procedural knowledge (e.g, [35], [26]; see also [12], [17]).

[3] For more details and a short critique of this approach see 6.2 on page 46.

ulated and mostly based on pattern retrieval. Formulated a few years later, on the basis of neurological findings, systems theory explains this distinction in terms of two different memory structures. Implicit memory is unconscious and does not decay with time, while explicit memory is conscious and degrades over time [25]. Sun has formulated the *dual representation hypothesis* ([30], [27]) which can be seen as an attempt to synthesize the processing and the systems theory. This hypothesis postulates that implicit-explicit differences are structural in nature, and come from different ways in which knowledge is stored and manipulated in the mind. At the *top level*, equivalent to explicit processing, the knowledge is stored in crisp, symbolic manner. Such knowledge is directly accessible and can be manipulated during the process of reasoning. In contrast, the implicit *bottom level* stores knowledge in a distributed manner, which is not directly accessible to the reasoning processes, but is capable of taking into account a much wider range of variables than can be consciously examined. Due to different internal representations, different learning mechanisms are at work at each level. The mind's performance during skill learning is a combination of the contributions of those two levels, and the degree to which each level contibutes can be manipulated. Finally, the two levels can exchange knowledge and work in synergy. When there is no explicit knowledge available, bottom-up learning dominates the skill learning process.

The next chapter presents a computational model based on this hypothesis, and sets up the stage for experiments which were conducted in order to verify the model.

Chapter 2

Problem statement

2.1 The CLARION model overview

Based on the *dual representation hypothesis*, a model is presented here which takes into account the distinction between the implicit and explicit learning, and which makes it possible to simulate both top-down and bottom-up learning on a single architecture. The model's name is CLARION, which stands for *Connectionist Learning with Adaptive Rule Induction ON-line*, and which was first presented by Ron Sun in 1999 [30]. This model is in the following chapters used to simulate two studies of the interaction of implicit and explicit learning which involved human subjects. The goal of simulationg human experiments is to provide evidence that the model, and its underlying hypothesis can adequately explain the distinction between implicit and explicit learning.

A natural choice for a model to explain duality of learning is a hybrid architecture with each type of learning represented by a different machine learning strategy. In CLARION, implicit learning is represented by a neural network trained using a modified Q-learning algorithm, and explicit processes are accounted for by a rule induction system. CLARION takes the environment state as its input, and outputs an action to undertake. The output is determined stochastically based on combined recommendations of the two systems. Figure 2.1 summarizes the CLARION architecture. The following sections describe the model in further detail.

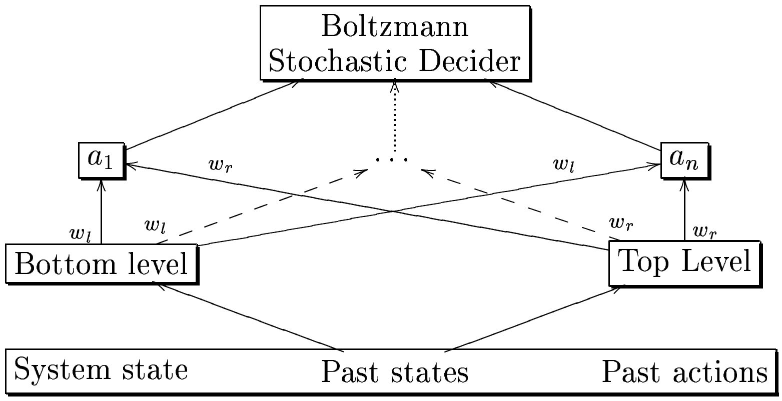

Figure 2.1: CLARION architecture. Possible system actions are $a_1 \ldots a_n$, and weights of lower and upper level recommendations are w_l and w_r respectively.

2.2 The bottom level of CLARION

A backpropagation neural network is a good choice to represent the implicit part of the mind. This is because network nodes are capable of working together, but are subsymbolic and not meaningful in separation from the network context. In fact this difficulty in access to explicit knowledge is neural networks' main weakness and makes it difficult to interpret results from applications where the nets are employed [23].

Whenever direct input-output mapping is available, the network learns with straight backpropagation. However, in the experiments discussed, the desired output was not available, but rather a feedback was offered for chosen actions. Often times the feedback would be delayed and only available when the subject ultimately achieved or failed to achieve the goal. For such cases, it is better to modify backpropagation, to include Q-learning [34].

Q-learning is an action-oriented reinforcement learning algorithm, where the learning agent maintains an internal approximation of the outside world in the form of values of each possible action a, at a given world state x. An action's value, Q(a —:x), represents the agent's expectation of a reward for taking action a in situation x. An agent chooses an action, observes the outside world, then chooses another action. When feedback becomes available, Q-values are adjusted according to the following formula:

$$Q(a| : x) = Q(a| : x) + \alpha \cdot (r + \lambda \cdot max_i Q(a_i| : y)) \tag{2.1}$$

where y is the next state resulting from choosing action a at state x, λ is the discount factor, r is reward at the next step, and α is the temporal discount factor. Q-learning is based on temporal difference. When reward is received, it is propagated backwards through all the steps taken leading to the reception of the reward, becoming smaller and smaller at each step. Q-value of an action a in state x is then the estimate at an expected reward down the line.

While agent's knowledge could be stored in a crisp manner as *state, action, Q-value* triples, it is often more advantageous to use a neural network. The number of possible states could be very large, and the agent would be at a loss at estimating the Q-values when a completely new state was encountered. If a neural network was set up to take x as its input, and output Q-values for each possible action, the agent could take advantage of the network's ability to geenralize between two similar states x and x'. Under such setup, the output error for backpropagation could be defined as $\delta Q(a| : x)$ at node representing the chosen action a, and 0 otherwise.[1]

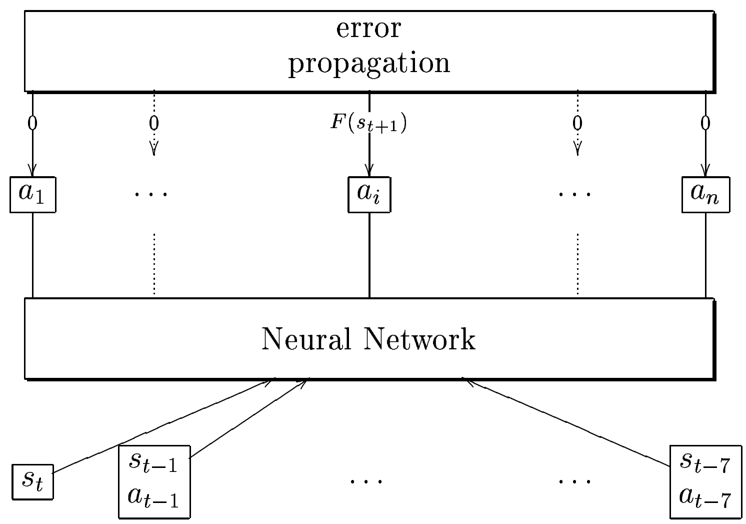

Figure 2.2: The bottom level setup. At step t, action a_i is chosen, out of n possible actions, resulting in system state s_{t+1}.

Figure 2.2 illustrates the way the neural network was set up. The input consisted of the current system state, and past state-action pairs for seven

[1]Alternate network setup is possible: input is x and an action a_i, and output is the action's Q-value. However, such setup would make it necessary to conduct forward propagation to assess each action, and making it computationally inefficient.

immediately preceeding time steps, where action $a_{t'}$ resulted in system state $s_{t'}$. The number 7 was chosen based on psychological data on attention span in humans [18]. Each action or state was represented in the input level by n binary neurons, with only one of them having activation value 1, and the rest set to 0. [2] Such input was fed to the hidden layer, and the output consisted of n nodes, representing the network's idea of q-values of each possible action. When CLARION chose an action and received feedback, only the error for this action was propagated back (the error terms for all other actions was 0). The error term for the chosen action a_i was a combination of feedback parameter and the activation value of the node:

$$Err(a_i) = \begin{cases} F \times Q(a_i) & \text{if } a_i \text{ was chosen} \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

and the feedback F was determined as follows:

$$F(s_i) = \begin{cases} F^+ & \text{if } s_i \text{ was a target state} \\ F_0 & \text{if } s_{i+1} \text{ or } s_{i-1} \text{ was a target state} \\ F^- & \text{otherwise} \end{cases} \tag{2.3}$$

where F^+, F_0, and F^- are respectively: positive, neutral and negative feedback parameters.

This is a simplified version of Watkins's Q-learning algorithm, applicable when feedback is available immediately after an action is chosen.

2.3 The top level of CLARION

In general, the top level captures explicit knowledge through a symbolic or localist representation, where each unit is immediately accessible for retrieval and has a clear conceptual meaning. Sun has developed a sophisticated general purpose *Rule Extraction and Refinement* algorithm [27], but in the simulations presented here, a simpler version, with hand wired rules was used. By criterion of robustness, this choice could be construed as a weakness of the simulations, so future work should examine whether CLARION can simulate process control task learning without the use of hand wired rules.

[2] While neural networks can learn to recognize more compact representations (for example, binary numbers, an isolated node representation chosen here should make the learning much faster, and have the added benefit of making the learned network weights easier to interpret for humans.

generation	rule format
1	$P_t = aW_t +$ b
2	$P_t = aW_{t-1} +$ b
3	$P_t = aW_t + cP_{t-1}$
4	$P_t = aW_{t-1} + bP_{t-2}$

Table 2.1: The order of rules to be tested. For each generation, a set of rules gets generated, each with unique values of parameters taken from the following sets: a = [1, 2], b = [-2, -1, 0, 1, 2], c = [-2, -1, 1, 2]. P is the system output, W is the user input, and parameter t signifies time step, with t-1 being the previous time step, and t-2 being the step before previous.

Table 2.1 summarizes the way the rule system was set up in these simulations. The hand wired rules are the agent's numeric hypotheses about which variables and to what extent are relevant in the next output of the system. Based on each such hypothesis, an agent can then determine its optimal action to produce desired system input, much like in equation 3.2 on page 23.

There are four generations of such rules, each generation concentrating only on certain variables. Within each generation the rules differ by degree to which each variable is responsible for the system output. Only one generation of rules is active at a time. As the agent chooses actions, and observes the outcomes, it can determine which rules are flawed, and which lead to favorable outcomes. After certain number of trials, the agent removes the rules which were determined to be flawed from the list of valid rules. If all rules from a given generation get deleted, the pool of current rules is populated by rules from the next generation. Each generation contains a more complex hypothesis form than the previous one, and the general form remains in accordance to the relations used in the human experiments which were simulated. The rules have been set up in such a way, so that one of the generations contains the correct rule whose application would lead to optimal performance. Normally, once the generation containing the correct rule is put into the pool at least the correct rule will survive the deletion process, thus halting the development of next generations. Due to random noise elements in the human experiments simulated, in the unlikely event that the optimal rule does not perform well, and all the rules in all four generations get deleted, the rule generation process would start again beginning with generation 1.

In order to determine a rule's usefulness, a variation of information gain measure was used. For a given rule C, its information gain value was:

$$IG(C) = \frac{PM(C) + 1}{PM(C) + NM(C) + 2} \tag{2.4}$$

where PM(C) is the number of instances where an action recommended by rule C was chosen and that action led to a reward, and NM(C) is the number of instances where such action led to undesirable consequences. After the number of applications, *PM(C) + NM(C)*, was greater than the minimum number of applications, r_n, and the rule's information gain measure fell below a threshold r_t, the rule was deleted.

2.4 Combining the contribution of each level to determine the model output

Each possible action has a separate corresponding intermediate node, connected to each rule in the top level, as well as its corresponding output unit in the bottom level. At the top level, a rule can recommend at most one action, with a constant recommendation strength of 1. This strength is the multiplied by the overall weight of the top level recommendations, W_r, and then added to the appropriate action node. This procedure is repeated for all rules. On the bottom level, after being multiplied by the corresponding lower level recommendation weight, W_l, each unit's activation is added on to its corresponding action output unit.

After each level's contribution to each action recommendation is credited, each intermediate action node, contains a numeric value which is a result of weighted contributions of each level towards this action. This value can be computed by the following formula:

$$Q'(a_i|x) = \overbrace{W_l \times l(a_i)}^{\text{bottom level}} + \overbrace{W_r \times \sum_j r_j(a_i|x)}^{\text{top level}} \tag{2.5}$$

where

- i is the number of possible actions,
- a_i is the i-th action,

- $l(a_i)$ is a bottom level's node activation corresponsing to action a_i,
- j is the number of rules at the top level,
- $r_j(a_i|x)$ is the j-th rule's binary (0 or 1) recommendation for taking action a in situation x

The actual action to be taken by the system is determined stochastically, according to the following formula fro the probability of a given action a in situation x:

$$p(a|x) = \frac{e^{Q'(a|x)/\alpha}}{\sum_i e^{Q'(a_i|x)/\alpha}} \tag{2.6}$$

where α is the temperature controlling the randomness of the decisions taken, and e is the natural log base. This formula is known as Boltzmann's sum, or Luce's choice axiom. This formula has its source in ideal gas physics, but in 1959 it has been introduced into behavioral science by Luce and has also proven successful in Q-learning [16], [34].

To simplify the discussion, the relation between W_r and W_l is established as follows:

$$W_r = 1 - W_l \tag{2.7}$$

where W_l is a real number between 0 and 1 (inclusive). [3] This is done without any loss of generality, since the value of p($a|x$) can remain the same by adjusting the temperature (see Appendix B.1).

Some general remarks about CLARION are in order. First of all, CLARION is directly based on the *dual representation hypothesis*, although it is obviously not the only possible model which could have this hypothesis at its base. Further, there are many versions of CLARION used in different simulations. The version presented here was specifically tailored to simulate the production control tasks.

2.5 Process control tasks

Process control tasks are one more tool which can be used to study bottom-up learning. Subjects are asked to interact with the system over a period of several trials, and on each trial the output of the system depends on the previous interactions and their results. The rules governing the system's

[3] And so is W_r, of course.

behavior are not disclosed to the subjects, and the subjects are then asked to achieve and maintain certain system state.

Process control tasks, also known as dynamic decision tasks, represent common problems encountered by people in everyday life. Examples of such problems are managing production processes, stock trading, driving in a strange city without a map, and many others. There are four characteristics which distinguish those tasks from single decision tasks, such as decision to have a baby, or decision to buy a new car:

1. A series of decisions is required;

2. Decisions are interdependent;

3. The state of the system changes autonomously, as well as a result of subject's decisions;

4. Decisions are goal directed.

Because of the real-time nature of the task, no single input-output mapping exists which can be used to achieve desired results. Even when simple underlying rules are used to govern the system's performance, human subjects require a significant time to discover them, and thus they have to initially rely on implicit processes to perform in a given task. With time, subjects develop explicit knowledge about the nature of the task, but their performance is improved even before the explicit knowledge is available. Since implicit processes and their knowledge are responsible for the initial performance gain, the direction of knowledge gain is bottom-up.

The particular variations of the process control tasks which were simulated here have been first proposed by Broadbent and Berry in 1984 [5]. They involve 12 possible inputs and 12 possible system outputs. There are two variations of the tasks: factory production and personal interaction. In the factory production task, subjects are asked to imagine that they are in charge of a sugar production factory, given control over the size of the factory's workforce, and then they are asked to maintain certain level of production. Factory workforce can range in whole hundreds of workers from 1 to 12, and the production level is given in thousands of tons, also ranging from 1 to 12. In the personal interaction task subjects are asked to interact with a computer person by selecting an abstract behavior level in reaction to the computer's own behavior level. There are 12 possible levels of behavior:

very rude, rude, very cool, cool, indifferent, polite, very polite, friendly, very friendly, affectionate, very affectionate, and loving. The subjects were asked to achieve and maintain certain level of behavior in the computer person.

The following two chapters show how CLARION was used to simulate certain process control experiments. The model's feasibility to represent bottom-up learning is demonstrated, its behavior during simulation is analyzed, and finally advantages and limitations of the model are discussed. A number of important process control experiments from the work of Stanley et al, and Berry and Broadbent was chosen for simulation.

Chapter 3

Simulating Stanley, et al

3.1 Background

In their 1989 paper, Stanley, Mathews and Buss examined dissociation between task performance and declarative knowledge in the process control tasks. [1]

Four experiments were conducted in which various aspects of bottom-up and top-down learning were examined. In experiment 1, subjects were asked to explain their actions during the experiment, thus enhancing the degree of bottom-up passing of knowledge. In experiment 2 explanations from the previous experiment's top performers were given to novice subjects to test under which conditions it was possible to pass knowledge on the top level. Experiment 3 examined varous effects on verbal training on performance in the task. Experiment 4 conducted additional trials to see whether it was possible to identify a point where the subjects' performance has significantly improved, and whether reaching of such insight into the nature of the system would result in substantive increase in declarative knowledge.

Stanley et al. used both versions of the process control tasks: personal interaction, and sugar factory. In both, the target level was set right in the middle - at 6,000 tons or "polite." For details on the workings of those tasks see 2.5. Results from both groups from Experiment 1 and two groups from Experiment 3 were chosen for simulation by the model. Other groups and experiments were not simulated for the reasons outlined below.

Experiment 2 dealt with passing of declarative knowledge to other sub-

[1] The human data in this chapter is based on [26]

jects. The finding of this experiment was that only the declarative knowledge gained in the last stages of the experiment by some subjects contained valuable insights into the system's workings. This kind of behavior is an integral part of the model - as more examples are presented, weak rules get weeded out, and only successful ones remain. Therefore there was no need to simulate this experiment.

Experiment 4 tested the subjects over additional trials to further examine at which point declarative knowledge begins to occur, and whether this point has correlation with actual performance improvement (researchers found out that performance is improved before subjects are able to verbalize useful knowledge about the task). Because the top-level rules were hand-coded in the model, it is dubious whether successful simulation of this experiment could lead to any general conclusions about the model's usefulness.

Experiment 3 examined the effect which various forms of verbal training have on subject performance. Performance of all the groups in all the conditions of this experiment was better than that of control group, but for groups in conditions which received very specific clues on how to approach the problem performance was worse than if those clues were followed to the letter by the subjects. The main finding of this experiment was that even though some verbal training methods offered significantly better degree of declarative knowledge than others, they all resulted in similar performance during the experiment (see 3.3 for discussion on the relevance of this finding). There were four conditions in this experiment, and two of them were chosen for the simulation. In order to simulate the other two, an extensive manipulation of the top-level would have to be performed, even though performance would be expected to remain the same. As previously mentioned, because the model's rules at the top level were hand-coded, any further manipulation could raise questions as to the usefulness of the simulation, and so it was decided to not simulate those conditions. The remaining two conditions from Experiment 3 are discussed in 3.2.2 together with the two conditions from Experiment 1, since there is no reason to discuss them separately.

3.2 Experimental setup

3.2.1 The task

Equation 3.1 was used to determine the system's response to user input.

$$s_{t+1} = (2 \times a_t) - s_t + N_t \tag{3.1}$$

where s_{t+1} is the next system state, s_t is the current system state, a_t is the user input, and N_t is a noise term which can take on a random value of -1,0, and +1 in a particular time slot t.

Despite the simplicity of the equation used in determining the next system state, human subjects during experiments have found the task to be quite difficult. For one, two interdependent variables, current input W_t and previous system state P_{t-1}, need to be taken into account. Further, the number of trials during the experiments discussed in this work was limited. Finally, the effect of noise does not allow for any hypothetical formula to be tested with 100 percent certainty.

Because of the noise factor, results within 1 below or 1 above were considered on target [2]. Even so, knowledge of the formula and its perfect application would result only in correct response on average only in 5 out of 6 trials. If the subject were to use formula 3.1 to determine the desired input before the noise term N_t was factored in, he would arrive at:

$$a_t = \frac{s_d + s_t}{2} \tag{3.2}$$

where s_d was the desired output. If the sum in the numerator, $s_d + s_t$, was even, then a_t determined from this formula would always result in an on-target system state, even after the noise term was factored in. Since s_d is constant, whether the sum is even depends solely on the previous system state, s_t. Since the highest possible system state s is even (12), and the lowest possible state is odd (1), s_t can be assumed to be even in exactly half of the cases on the average, which means that on average in half the cases it will be possible to get an on-target result with 100 percent certainty. A more interesting case arises in the other half of the cases, when the sum is odd. Since user input can only be given in whole numbers, a_t has to be rounded. If one were to round it down, the next output would be 1 below the target level, s_d, before the noise term was added in. In such case a noise of +1 would result in the production level being off-target, while the remaining two possible noise values would give an on-target result. Equivalent reasoning

[2]It is unclear whether subjects were aware of this criterion. Since Stanley et al.'s formulation of the task followed that of Berry and Broadbent in [5], it is safe to assume that this criterion was also the same, and therefore the subjects were *not* aware of it.

can be applied in the case if a_t was rounded up (with noise of -1 causing the mischief). Since noise is random, on average in 1 out of 3 cases when the sum in the numerator was odd, the perfect application of the formula would give off-target result. Thus we can calculate the probability, Φ, of being on target in any random state:

$$\Phi_{ontarget} = \frac{1}{2} \times 1 + \frac{1}{2} \times \frac{2}{3} = \frac{5}{6}$$

3.2.2 Setup

In Experiment 1 there were two conditions. One was a control condition, where subjects were simply asked to score within the target range on either . The other was called "original" condition, where subjects were asked to record instructions for an unseen partner on how to best perform the task. It is refered to in further in this work that the subjects were asked "to verbalize." There were two further conditions in Experiment 3 which we chose to simulate. Under the memory training condition, prior to the test the subjects were given a chance to study 12 randomly generated correct examples in the form:

System state is Q. To get to target range, choose response X, which will result in state R. Then, to get to state S, choose Y.

In the simple rule condition of Experiment 3, subjects were told to always select a response half way between current state and target state. If this rule was adhered to perfectly, subjects could expect to perform as well as if they knew the exact formula used to determine the system state; i.e. they could expect to score on average 8.3 on target responses in a block of 10 trials. For each condition there were 2 groups of subjects, one being exclusively tested on the person interaction task, and the other working on the sugar factory task. The number of subjects in each group varied from 12 to 15, with original sugar task group numbering 31, and original person task group numbering 22.

Target level in all cases was right in the middle of the system response range: it was 6,000 tons, or "polite" respectively. A response within one level of the target range was considered on target. Subjects were tested over a set of 20 blocks, each block consisting of 10 trials. For each block the system state would be initialized randomly. Subject would then enter his chosen input, and the system would display its next state according to formula 3.1. This constituted a single trial, and after 10 such trials, the system state would be randomly reset, and the next block would begin.

3.2.3 Results

Table 3.2.3 summarizes the average scores in selected groups of Stanley et al. experiments. The scores are per subject on-target responses out of 10. This statistic makes more sense once experiment's context is taken into account, since each block consisted of 10 trials.

Group	Sugar task	Person task
control	1.97	2.85
original	2.57	3.75
memory training	4.63	5.33
simple rule	4.00	5.91

Table 3.1: Human data in Stanley et al. experiments (average on-target result out of 10 trials

The direct finding of Experiment 1, later confirmed in Experiment 3, was that verbal instructions lead to increased performance on the task. One surprising finding was that while scores on person task paralleled those on sugar factory task, they were consistently higher. Finally, performance gain in the original condition was much higher in the person task than in the sugar task.

3.3 Experimental results

Based on the verbalizations of subjects in some groups, Stanley et al. have also found that there was a dissociation between task performance and verbalizable knowledge. Subjects tended to become quite skilled at either task long before they could verbalize successful rules. This finding was in direct opposition to theories which viewed learning as a purely top-down process (i.e. [15]), and in fact are a strong argument for the existence of bottom-up learning. Stanley and his colleagues hypothesized the existence of two distinct cognitive structures although their insight into the nature of the two structures was different than that postulated in the CLARION model [3].

One counterintuitive finding of the experiments was that while performance was enhanced on all tested groups, no verbal training condition designed to enhance performance was better than others. Clearly, some of the advice given was better than others if applied by subjects indiscriminately, but it turned out that it was not what humans did in the experiments. Stanley et al. speculated that a theory based on classifier systems developed by Holland could explain this phenomenon. According to this theory various sets of rules compete for control of behavior [10]. Rules which perform better become strenghtened after each use, ones which are unsuccessful lose strength. Stanley hypothesized that the rules given in various experimental groups in order to faciliate performance had to compete against other rules which were based on subjects' previous experiences. It should be pointed out that the CLARION architecture is also based on the same assertion.

Another surprising finding was significantly better performance on the person task than on sugar factory task accross all groups of subjects. Both of those tasks were based on identical rules, had the same internal mechanisms involved, and yet something about the presentation of the problem as personal interaction versus sugar factory production has made it easier for subjects to learn. Stanley and colleagues explained this by noting that, due to their life experiences, subjects are more likely to be familiar with personal interaction setting, and therefore they have preconceived notions about how the system should operate, and also have an easier time recalling passed trials. Qualitative analysis of some of the subjects' verbalizations seemed to confirm that explanation.

3.4 CLARION simulation

3.4.1 Model setup

As noted earlier, Stanley and colleagues hypothesized that the improved person task performance was due to the better familiarity with this general type of setting by the subjects. Pre-training the model with 10 blocks for the

[3] It should be pointed out that Stanley et al. were not using the "top-down" and "bottom-up" terminology. This concept is of later origin. Stanley et al. thought the two structures to be a mental model of the task, and memory experiences associated with the task.

total of 100 trials, was sufficient to produce the same effect in the simulation.

For a general model description, see 2.1. In this simulation, temperature in the action choosing module was 0.01. The rule deletion threshold was set at 0.15 in the Control Group simulation. In the other groups' simulations, this threshold was raised to 0.35. This was done to capture the verbalization condition; to model the likely human subject reaction, model instances were "paying more attention" to critically evaluating various top level hypotheses. In the Memory Training Group simulation, 12 random correct examples were generated, and they were each wired into the top level as simple rules in the form *if* s_t *then choose* a_t.[4] To simulate the simple rule condition, the heuristic rule, as given by experimenters to the subjects, was wired into the top level.

The number of subjects in each group of the original experiment was 12 to 31. Considering properties of computer random number generators, and given enough simulation runs, for small number of subject in any group, one could find a simulator run to fit any preconceived data. In order to minimize the influence of chance factors in the simulation, each group had 1,000 simulated subjects.

3.4.2 The match

Table 3.4.2 summarizes the average scores in selected groups simulated by the model. The scores are in the same format as Stanley et al. used in their report (per subject on-target responses out of 10).

Group	Sugar task	Person task
control	2.28	2.61
original	2.95	4.19
memory training	4.09	5.43
simple rule	4.07	5.07

Table 3.2: Model data in the simulation of Stanley et al. experiments (average on-target result out of 10 trials

The simulation captured the verbalization the effect in the human data well. A t test was used to compare the Original Group with the Control Group in the model data, and it showed a significant improvement of the

[4] Actual examples in the human experiments involved a time-series of three action-result sequences.

former over the latter ($p < 0.01$), matching the improvement shown in the human data. The simulation also captured the explicit instruction effect. Pairwise t tests were used to compare Memory Training and Simple Rule groups with Control Group in the model data, and they showed significant improvements in these two groups' performance over Control Group ($p < 0.01$).

Chapter 4

Simulating Berry and Broadbent

4.1 Background

In their experiments on human skill learning in process control tasks, Stanley and colleagues noted unexpectedly better performance accross all experimental conditions in subjects faced with personal interaction task, than subjects working on the sugar factory production task, even though those tasks are essentially the same [26]. Stanley et al. speculated that this was due to the fact that subjects had more prior experience in personal interaction than in factory environment, and therefore were able to form commonsense assumptions about the task, and also had an easier time recalling examples in the experiment by being able to put them in familiar perspective.

When subjects of the Stanley et al. experiments were asked to verbalize [1], the performance gain was proportionally much higher in the personal interaction task than in the sugar factory task. In essence, not only was the initial performance better in person task, but the subjects also learned the skill faster. Something about the person task, once again possibly prior familiarity, allowed for faster learning.

Adopting the dual representation hypothesis can give further insight into the dynamics of this learning phenomenon. Under the dual representation hypothesis, verbalization condition in Experiment 1 of Stanley et al. could be represented as giving higher precedence to recommendations given by the

[1] "Original" training condition in Experiment 1 in [26]

top level. CLARION simulation in Chapter 3 which used this interpretation, successfuly matched human data. Dual representation hypothesis explains that if a mind gives higher precedence to recommendations of the top level, then those recommendations will be more effective if successful task performance involves application of common-sense rules which the top level already knows, and less effective if the rules are less commonsense.

Performance gains in the person and sugar factory task of the original group in Experiment 1 of Stanley et al. are good evidence for the dual representation hypothesis, but they are insufficient, since they are based on a speculation that person task appears more common-sense to the subjects than sugar factory task. One year earlier, Berry and Broadbent also investigated this area, and managed to create conditions under which such speculation was not necessary. In addition to some surprising findings, the results of their experiments have provided even better evidence for the dual representation hypothesis.

Berry and Broadbent, in their 1988 paper reported two experiments. [2] The first one examined the effect that asking subjects to identify the pattern in system's responses had on two different versions of the personal interaction task. The first version of the task was constructed in such a way as to make it easier to figure out, while the second version was more difficult. The second experiment examined how subjects who have attained some familiarity with the personal interaction task would do on an identical task whose superficial setting was different. Transfer of knowledge and skill between different tasks is outside of the scope of this work, and so the second experiment has not been simulated here.

Even though their experiments served to give an insight into some of the puzzling Stanley results, Berry and Broadbent's inspiration lied elsewhere. They were inspired by work on artificial grammar learning task [20] conducted by Reber and colleagues. Reber et al. have asked subjects to memorize a sample of strings which did not represent any human language, but were generated in accordance with artificial, arbitrary rules. Then, without warning, the subjects were shown a series of other strings, and asked to identify which strings obeyed the same rules as the ones which were being memorized awhile ago. One set of subjects would receive explicit instructions to try and discover patterns behind the strings, others remained oblivious to the existence of any rules behind the string generation and served as control.

[2] The human data in this chapter is based on [6]

One of the results of this work was to confirm the existence of implicit learning, since even the control subjects performed well above chance on the discrimination task. Further, Reber and his colleagues have demonstrated that explicit search instructions would only give better performance if the original strings for memorization were presented in a structured manner which made some patterns apparent. Otherwise, explicit search group performed worse than the control group.

The degree to which patterns in the presented material were obvious to the subjects in experiments of Reber and colleagues had important impact on performance. Reber et al. coined the term structural salience to describe such level of awareness. More generally, salience is the degree to which patterns underlying a learning task are obvious to the subjects. In Experiment 1 of their 1988 work, Berry and Broadbent have investigated how salience influences performance in personal interaction version of the process control task.

4.2 Experimental setup

In Experiment 1, subjects were faced with standard version of the personal interaction task. For more information on this task, see section 2.5. The system output for the salient condition was given by the following formula:

$$s_{t+1} = a_t - 2 + N_t \tag{4.1}$$

where s_{t+1} is the next system state, a_t is the user input, and N_t is a noise term which can take on a random value of -1,0, and +1 in a particulat time slot t. A different formula, given below, determined output in the non-salient condition:

$$s_{t+1} = a_{t-1} - 2 + N_t \tag{4.2}$$

where a_{t-1} is user input at the previous step. Berry and Broadbent assumed that introducing such response lag will make the task more difficult to figure out to the subjects.

The four groups were salient control, salient experimental, non-salient control, and non-salient experimental. Subjects were tested in 3 sets of 20 trials. Each trial consisted of a single input and system response. The system was in state 6 ("polite") at the beginning of each set, and subjects were asked to reach state 9 ("very friendly"). Results within one of the target range were considered on target to account for the random element, but the subjects were not aware of this criterion. Note that in this experiment, unlike in Stanley's, knowledge of the underlying formula would allow one to score on target every time. One exception would be the first trial in the non-salient condition, where the system response would be random, since there was no previous input to go on. For more details on the performance predictions in Stanley et al. see section 3.2.1. After the first set of trials, subjects in the experimental groups were asked to try and discover the pattern behind the system's responses, and were told that the pattern depended on their own inputs in some way. After the second set of trials, subjects in the experimental group were told which was the critical input step in determining system's response; in the salient condition this was the immediaely preceeding input, in the non-salient one, it was input from the previous trial.

4.3 Experimental results

Figure 4.1 depicts the results of the human experiments in graphical form. Numeric values are given in Appendix A.

The significance of those experiments was tri-fold. First of all, they demonstrated that whenever immediate application of commonsense reasoning is likely to fail (non-salient condition), performance will drop if humans are prompted to rely on explicit learning. Second, it was demonstrated that explicit learning was superior to implicit learning when it was possible to come up with useful rules to guide performance. It should also be noted that the results of the Berry and Broadbent experiments parallelled those of Reber et al. in [20] in artificial grammar learning. Since the two tasks are much different, it could be argued that the results could be interpreted to indicate more general properties of human learning, and similar results could be obtained on any skill learning task. Further, experimental results can serve to interpret the increased learning rate on the person task in Stanley et al. As is apparent from the figure, learning progresses much faster for salient tasks, and person version of the process control task in Stanley experiments was likely more salient than sugar factory version. Further, based on post-task questionaires, Berry and Broadbent have noted that explicit learning is used even in the absence of verbal instruction form the experimenters, it is just so to a lesser extent than in the verbal instruction condition.

Berry and Broadbent attributed performance drop in the explicit search

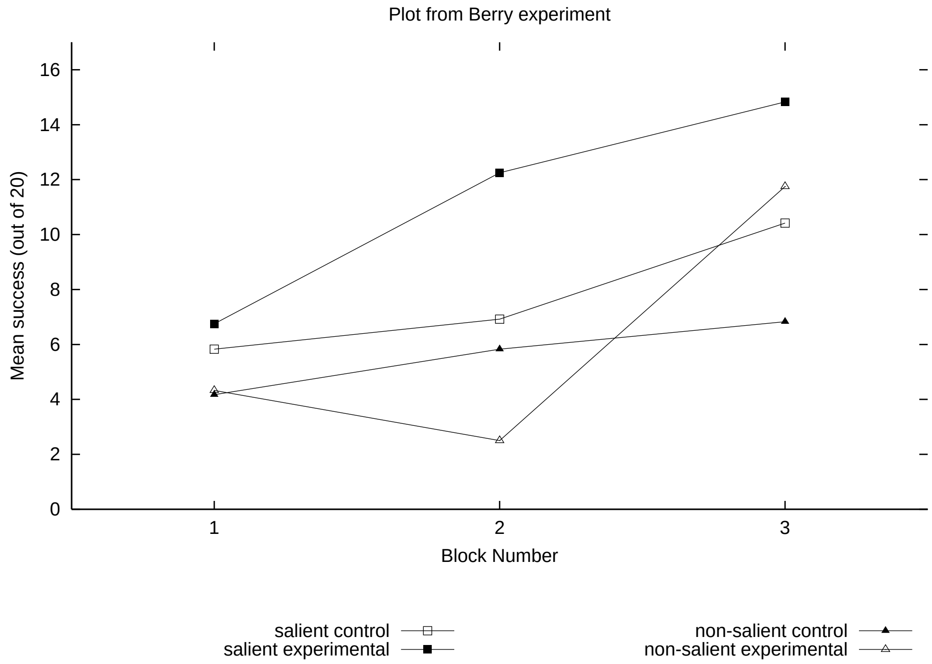

Figure 4.1: Broadbent and Berry experiment results in graphical form.

condition of the non-salient group to phenomenon they called selective learning. In implicit mode, subjects would build up a large number of variable contingencies and test them in parallel. The learning would progress slower, but ultimately had the chance of capturing non-salient variable relationships, and allowing for improved performance. When prompted to search for patterns, subjects would limit the number of variables considered for contingencies, thus reducing the time it took to learn. However, if the crucial variables were excluded from consideration, such learning would fail.

This explanation does not fully account for the phenomenon of inability to produce useful task advice, even when effective implicit learning took place. If implicit learning used the same mechanism as explicit learning, one would expect the knowledge of relevant contingencies and variables to be available from both levels once learning was completed. However, Stanley et al. have noted that knowledge from implicit level was not readily available. [26] Berry and Broadbent stated that implicit knowledge may be harder to recall, because of the difficulty of choosing among the many established relationships, or having low confidence in any particular relationship.

This explanation does not seem as convincing as the alternative presented here. The dual representation hypothesis, as presented here, explains this phenomenon much better, by allowing implicit knowledge to be stored in a distributed, continuous manner, which does not yield readily to retrieval and interpretation, while explicit knowledge is stored in the form of crisp rules.

4.4 CLARION simulation

4.4.1 Model setup

For a general model description, see section 2.1. In this simulation, the rule deletion threshold was set at 0.1 initially. To capture the explicit search instruction during the second training set, the rule deletion deletion threshold was raised to 0.5, and only salient rules were hypothesized in block 2. This was to model the increased learning ativities in the top level, and selective learning. The weight of each level's recommendation was changed to 0.5 to account for more reliance on the top level. To capture the relevant variable instructions given during the third training set, only rules that related the given critical variable to the system output were hypothesized and tested at the top level (that is, W_t in the salient version, and W_{t-1} in the non-salient version). The learning rate was 0.04. The temperature in the action choosing module was 0.01. The momentum was 0.04. In order to minimize chance factors in the simulation, each simulated group had 1,000 subjects. For more explanation on why the number of subjects simulated was much higher than in the original experiments, see section 3.4.1 on page 27.

4.4.2 The match

Figure 4.2 summarizes the simulation results in graphical form. Numeric values are given in Appendix A.

The model has succeeded in simulating the performance drop in the explicit search instruction of the non-salient condition (Step 2). The non-salient experimental performance drop in step 2 was not as dramatic as in the human experiments, but nevertheless it was significant, considering that the model's results are most likely the asymptotic limits as the number of subjects approaches infinity. The model was also successful at simulating the dramatic performance increase after the relevant variable was given to the subjects

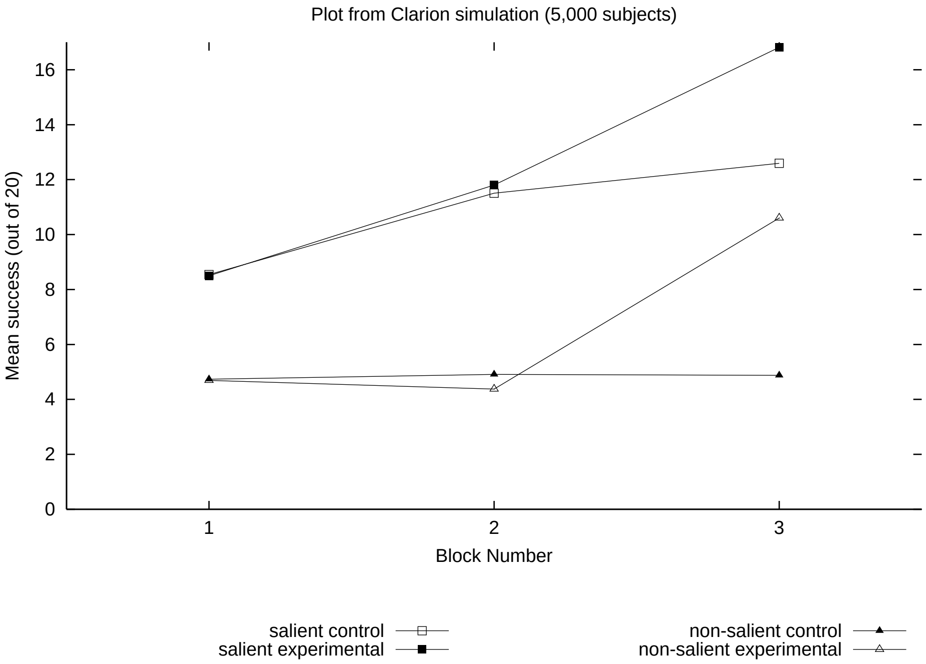

Figure 4.2: Simulation results in graphical form.

(Step 3). Performance raise in the non-salient group was not as high as to surpass the non-salient control group's in Step 3, as was the case in the human experiments, but nevertheless it was substantial, and both experiemntal groups diverged dramatically from their respective control groups.

Chapter 5

Optimizing the simulator

5.1 Motivation

Simulating human experiments in Chapters 3 and 4 involved identifying which model's parameters corresponded to experimental conditions the tuning parameters to obtain results matching human data as closely as possible. The model has twelve free parameters which are summarized in Table 5.1. Since some of those parameters are integers, and some are real numbers, it

Top Level	
r_tO	initial rule deletion threshold
r_tS	rule deletion threshold in explicit search condition
r_n	rule application minimum
Bottom Level	
M	momentum
λ	learning rate
N_{hidden}	number of hidden network units
F^+	positive feedback
F_0	neutral feedback
F^-	negative feedback
Other	
W_lO	initial weight of lower level recommendations [1]
W_lS	weight of lower level recommendations in the modified group
α	temperature in the decision-making process

Table 5.1: Relevant CLARION free parameters

is difficult to gauge the complexity of the optimization task. One way of optimizing model performance is hand-tuning of all of the parameters, based on the assumption of near linear sensitivity, knowledge of which parameter settings should make sense in given experimental condition to be simulated, and subjective experience with the model. Hand-tuning has proven successful in simulating the Stanley experiments in Chapter 3, but it is not difficult to notice that the results of the simulations in Chapter 4 do not match data very closely. While the major trends were captured, there is room for improvement. Based on experience with hand-tuning the model, the number of possibilities is too large to be adequately explored by adjusting the parameters manually. In this chapter, a different approach is taken. Without presuppositions as to which parameters are responsible for which experimental setting, a search is performed for parameters which match the human data as closely as possible.

Before discussion on the details of the optimization process the matter of automatic optimization needs to be examined. Automatic optimization of cognitive models could be criticized by those who are afraid of leaving the cognitive context in the pursuit of optimal match results. The new found parameters might not make sense in the context of the experiment, or they might not be general enough to be applicable in other experiments. However, this argument can be refuted. If intelligent mind has evolved over time, then evolution itself can be viewed as computational process of optimization, and therefore exploring model optimization can be justified in cognitive context. Exploration of free parameter space can also provide further support for the model: whether it is robust and how it behaves under extreme conditions, what type of solutions it converges to, and whether some of these solutions could correspond to the states found in nature.

Additional motivation comes from Criterion 3 in section 6.1 (see page 45). It would be desirable if the model could robustly hand-tune its own parameters within reasonable time, to match some arbitrary experimental data.

Therefore, it was decided to try one of the automated non-linear optimization methods to search for parameters which would allow to match the human data in the Broadbent and Berry experiments more closely.

5.2 Optimization details

Table 5.2 summarizes the range of parameters which was considered during the optimization process. There are 12 parameters, making for a search

Symbol	Range
r_tO	0.0001 to 0.9999
r_tS	0.0001 to 0.9999
r_n	1 to 20
M	0.0001 to 0.9999
λ	0.0001 to 0.9999
N_{hidden}	10 to 200
F^+	-1.00 to +1.00
F_0	-1.00 to +1.00
F^-	-1.00 to +1.00
W_lO	0.0001 to 0.9999
W_lS	0.0001 to 0.9999
α	0.0001 to 0.9999

Table 5.2: Sensible range of values for CLARION parameters

in 12-dimensional solution space. There are several choices as to the error function. One is a simple euclidean distance between each of the model's results and human experiment results. If we allow scaling of the numeric results, another possibility is to solve a set of linear equations to find such constant with which to multiply the experimental result vector for which the euclidean distance would be smallest, and then use this smallest distance as the error value. [2]

Some of the parameters are discrete, which might present problems to some optimization methods, such as gradient descent. Also, in order to compute the error function each time, it is necessary to run the full simulation, which takes considerable time. This constraint makes it impractical to use genetic algorithms, which require numerous evaluations of the error function. After taking these considerations into account, scatter search was chosen as the most suitable general method for model optimization. Scatter search has the additional merit of that it is simple to implement and highly robust.

[2] Finally, one could factor in desired trends such as one vector dimension should remain above another, but it has not been tried.

5.3 Optimization method

General optimization techniques, known as metahueristics, can be further subdivided into Adaptive Memory Programming [31]. All of the adaptive memory programming techniques share common characteristics:

1. a population of solutions is maintained;

2. provisional solution is constructed based on the past;

3. provisional solution is improved on;

4. the new solution is included into the past experiences.

Adaptive memory programming includes genetic algorithms, taboo search and scatter search, and combinations of hybrid techniques used between the three.

Scatter search was proposed by Glover in 1977 [8]. It is similar to genetic algorithms in that it maintains a population of solutions, but the difference lies in representation, mating, and lack of clearly defined generation concept. Implementations largely differ in details, but usually among the population of solutions two or more are chosen to produce an offspring. The offspring shares a linear combination of its parents' characteristics. The offspring's characteristics are then processed to make sure they are feasible, match constraints, and any other local improvement techniques are applied. Then the offspring is made to replace one of the solutions in the solution set.

OptQuest is a commercial product which uses a hybridized version of scatter search. It is designed for optimization of systems where it is important to make as few calls to the evaluation function as possible. Model optimization used here uses techniques described in OptQuest white paper by Laguna [13].

The reference set of currently best solutions is initialized by generating 100 random points-solutions in the 12-dimensional search space. This set includes a midpoint $x_i = l_i + (u_i - l_i)/2$, where u_i and l_i are the upper and lower bounds on x_i. Each new solution is subject to normalization, where its integer parameters are assured, bounds on all parameters are maintained, and then the solution is evaluated by the evaluation function. After initialization, the search repeatedly mates two selected parent solutions to produce 4 offspring, of which only one with the best value of the evaluation function will be placed in the reference set. The offspring from X and Y are generated as linear combinations of parents, according to the following formula:

$$\begin{aligned}P &= X + D \\ Q &= X - D \\ R &= Y + D \\ S &= Y - D\end{aligned}$$

where D = (X - Y)/3. The two parent points X and Y are chosen probabilistically, based on the value of their evaluation function, and how long they have been in the reference set. Longer reference set tenure makes it more likely to be selected for mating, which allows even solutions with very poor evaluation function value to be eventually considered. One additional factor is used in parent selection: solutions which were chosen in the past 7 matings cannot be selected again. Once the 4 offspring are evaluated, the best of them replaces the worse of the parents in the reference set.

5.4 Error functions

The set of 12 model results from different conditions and stages of the experiment to be matched to human data could be seen as a point in 12 dimensional space. One simle error function is a sum of errors on each of the trial runs. Another, related error function is to the euclidean distance between model and human results:

$$Error = D = \sqrt{\sum_{i=1}^{12} (x_i - y_i)^2} \tag{5.1}$$

where x_i and y_i are the ith model and human result respectively.

Much better match on trends to be simulated could be obtained if scaling of the results by some constant number was allowed. For each set of results $\{x_1, \ldots, x_{12}\}$, a constant a is computed such that the euclidean distance from $\{a \cdot x_1, \ldots, a \cdot x_{12}\}$ to the human experiment results is minimal. Then the scaled euclidean error function is given by:

$$Error_{scaled} = D = \sqrt{\sum_{i=1}^{12} (a \cdot x_i - y_i)^2} \tag{5.2}$$

Using a from equation B.2 in Appendix B.

5.5 Results

Tables 5.3, 5.4 and 5.5 summarize the sample results for each of the error functions.

Search parameters	
Initial population	100
Number of iterations	1,000
Error function	sum of errors
Hand tuning error	20.43
Best solution error	20.92
Best solution	
Model parameter	Value
r_tO	0.28
r_tS	0.08
r_n	7
M	0.22
λ	0.95
N_{hidden}	106
F^+	0.27
F_0	0.91
F^-	-0.63
W_lO	0.15
W_lS	0.37
α	0.25

Table 5.3: A solution found by Scatter Search using sum of errors error function.

None of the searches resulted in a solution matching human results better than hand-tuning. The search using the scaling error function was hindered, because parents were mated without taking each parent's scaling context into account. Another difficulty that the search has encountered was that the error function values obtained for simulation with 100 subjects, were not always as accurate as the asymptotically converging results for 1,000 subjects. It was necessary to limit the error function sampling to 100 subjects during the search because of computational time limits. There is much room for improvement in this area, since applying optimization to new areas usually involves much trial and error at the start.

Search parameters	
Initial population	100
Number of iterations	400
Error function	euclidean distance
Hand tuning error	7.08
Best solution error	9.70
Best solution	
Model parameter	Value
$r_t O$	0.62
$r_t S$	0.35
r_n	3
M	0.91
λ	0.58
N_{hidden}	96
F^+	0.21
F_0	0.38
F^-	-0.74
$W_l O$	0.76
$W_l S$	0.41
α	0.21

Table 5.4: A solution found by Scatter Search using euclidean distance error function.

Search parameters	
Initial population	70
Number of iterations	150
Error function	scaled euclidean
Hand tuning error	5.95
Best solution error	9.51
Best solution	
Model parameter	Value
r_tO	0.84
r_tS	0.98
r_n	4
M	0.76
λ	0.87
N_{hidden}	159
F^+	0.77
F_0	-0.09
F^-	-0.50
W_lO	0.40
W_lS	0.94
α	0.86

Table 5.5: A solution found by Scatter Search using scaled euclidean distance error function.

Hand-tuned solution	
Model parameter	Value
$r_t O$	0.1
$r_t S$	0.5
r_n	1
M	0
λ	0.04
N_{hidden}	40
F^+	0.5
F_0	0.3
F^-	-0.1
$W_l O$	0.8
$W_l S$	0.5
α	0.01

Table 5.6: A hand-tuned solution used in Chapter 4 given for comparison purposes.

On the other hand, some results are encouraging. First of all, the some searches were able to discover the commonsense feedback direction for the neural network, and the commonsense direction of change in lower and upper level contribution wieghts. Second, the entire population seemed to converge to the obtained error function values quickly while maintaining diversity in the parameter values. This tendency serves as evidence that the model is robust, and many parameter changes do not dramatically affect its performance.

Chapter 6

Conclusions

6.1 Criteria for success

In order to evaluate the model and the simulations in this work, the following criteria are established :

1. Is the model cognitively justified?
2. Can the model simulate human behavior adequately?
3. How robust is the model (how much tailoring is neccessary for a given task simulation)?

These criteria are not general criteria for all cognitive models, but are specific to how the focus of this work has been defined.

In terms of Critetion 1, Chapter 1 has tried to explain the motivation and cognitive justification for the *dual representation hypothesis* which CLARION is based on. For Criterion 3 CLARION can learn on-line, on a case-by case basis, which is a step in the direction of robustness. However, it should be noted that while the same general architecture has been applied to a wide number of problems, but it required some problem-specific tailoring, and in this case the rules pre-generated at the top level may see somewhat artificial. See Section 6.3 on page 46 for a discussion of this particular issue. The contribution of this work figures into addressing Criterion 2 by showing how CLARION can simulate a number of human experiments in process control tasks. Other works have shown CLARION's applicability to a wide range of human experiments as shown in section 6.3.

6.2 Comparison with ACT-R

Few other cognitive models so far have attempted to simulate the implicit-explicit learning. The most prominent of them, ACT-R is an intresting production system using connectionist notions as a mechanism for choosing actions [2]. By associating several parameters with each production rule, and then using these parameters in decision-making, this mechanism allows for forgetting of old rules and learning which rules are relevant in a given situation. Lebiere et al. have used this mechanism to emulate implicit learning in humans, while a search induced by the production rules themselves accounted for explicit learning [14]. ACT-R suffers from one weakness - it has no way of accounting for bottom-up learning. The only way new production rules can be generated is if they're encoded by hand, or if a rule is put forward to start a search. The connecionist mechnisms can turn certain rules on and off, but they cannot generate new knowledge. In contrast, CLARION, in principle, allows the implicit level to store knowledge, and allows to generate new rules at the top level by observing the results of application of bottom level's knowledge [27].

6.3 Future work

One obvious limitation which the model has was the use of hand-coded rules in the Rule Induction part of it. Ideally, one would want the model to learn something new from the task in a robust manner. However, such use of hand-coded rules is not uncommon in the cognitive modeling field. For example, Lebiere et al. used such rules in their simulation of Anderson's Tower of Hanoi experiments [3] [14]. It should also be pointed out that on the same simulation of Anderson's Tower of Hanoi task, Sun et al. were able to succeed with CLARION without hand-coding any rules at the top level [28].

One area which would require further investigation in the future comes from the fact that there was considerable difficulty in simulating performance drop in the search instruction condition in the non-salient set of the Berry and Broadbent experiments. It would seem like the model is inherently set on learning, and considerable steps have to be taken in order to faciliate performance drop once the learning has started to take place. This is one area where the model could be improved in the future.

Finally, not all of the Stanley experiments have been simulated. While reasons for such choice are addressed in 3.1, one might expect that a robust model would be able to simulate a wide range of tasks and experiments. Should this be the only study where CLARION has been applied, other researchers might remain skeptical about the model's usefulness. However CLARION has already been used to simulate a significant number of skill learning experiments with human subjects, on a number of different problems, including minefield navigation, and Tower of Hanoi tasks [29] [28].

6.4 Closing remarks

This work has presented a hybrid connectionist model which accounts for differences in explicit and implicit learning in humans. The model was used to simulate a number of key experiments which studied the dysociation between the two types of learning. Upon critique, the model has been shown to demonstrate much promise in explaining human learning and understanding the experimental results.

Bibliography

[1] J. R. Anderson, (1983). *The architecture of cognition.* Harvard University Press, Cambridge, MA.

[2] J.R. Anderson, (1990). *The adaptive character of thought.* Erlbaum, Hillsdale, NJ.

[3] J. R. Anderson, (1993). *Rules of the Mind.* Lawrence Erlbaum Associates. Hillsdale, NJ.

[4] J. Anderson and C. Lebiere, (1998). *The Atomic Components of Thought.* Lawrence Erlbaum Associates. Hillsdale, NJ.

[5] D. Berry and D. Broadbent, (1984). On the relationship between taks performance and associated verbalizable knoledge. *Quarterly Journal of Experimental Psychology.* 36A, 209-231.

[6] D. Berry and D. Broadbent, (1988). Interactive tasks and the implicit-explicit distinction. *British Journal of Psychology.* 79, 251-272.

[7] H. Dreyfus and S. Dreyfus, (1987). *Mind Over Machine: The Power of Human Intuition.* The Free Press, Glencoe, IL.

[8] F. Glover, (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences 8.* 156-166

[9] M. Heidegger, (1927/1962). *Being and time.* New York: Harper and Row. [English translation]

[10] J.H. Holland, K.J. Holyoak, R.E. Nisbett and P.R. Thagard (1986). *Induction, Processes of Inference, Learning and Discovery.* MIT Press, Cambridge, MA.

[11] W. James, (1890). *The principles of psychology.* New York: Dover.

[12] A. Karmiloff-Smith, (1986). From meta-processes to conscious access: evidence from children's metalinguistic and repair data. *Cognition.* 23. 95-147.

[13] M. Laguna, (1997-98). Optimization of Complex Systems with OptQuest. *OptQuest for Crystal Ball User Manual*, Decisioneering. Also available as white paper from *http://www.decisioneering.com/optquest/complexsystems.html.*

[14] C. Lebiere, D. Wallach, and N. Taatgen, (1998). Implicit and explicit learning in ACT-R. *Proc. of ECCM'98*, pp.183-189. Nottingham University Press.

[15] W. Lewis and Anderson, J. R. (1985). Discrimination of operator schemata in problem (...) Learning from examples. *Cognitive Psychology.* 17, 26-65.

[16] D. Luce (1959). *Individual Choice Behavior.* Wiley, New York.

[17] J. Mandler, (1992). How to build a baby. *Psychology Review.* 99, 4. 587-604.

[18] G. Miller, (1956). The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *The Psychological Review*, 1956, vol. 63, pp. 81-97. Also available online at: *http://www.well.com/user/smalin/miller.html*

[19] A. S. Reber (1967). Implicit learning of artificial grammars. *Journal of verbal learning and verbal behavior.* 5, 855-863.

[20] A. S. Reber, S. M. Kassin, , S. Lewis, and G. Cantor (1980). On the relationship between implicit and explicit modes of learning a complex rule structure. *Journal of Experimental Psychology: Human Learning and Memory.* 6, 492-502.

[21] H. K. Roediger, (1990). Implicit memory: retention without remembering. *American psychologist*, 45, 1043-1056.

[22] P. Rosenbloom, J. Laird, and A. Newell, (1993). *The SOAR papers: Research on Integrated Intelligence.* MIT Press, Cambridge, MA.

[23] D. Rumelhart, J. McClelland and the PDP Research Group, (1986). *Parallel Distributed Processing: Explorations in the Microstructures of Cognition.* MIT Press, Cambridge, MA.

[24] P. Smolenski, (1988). On the proper treatment of connectionism. *Behavioral and Brain Sciences.* 11, 1-43.

[25] L. R. Squire, B. J. Knowlton (1995). Memory, hippocampus, and brain systems. In M. S. Gazzaniga (Eds.) *The cognitive neurosciences.* MIT Press; Cambridge, MA. pp. 825-838

[26] W. Stanley, R. Mathews, R. Buss, and S. Kotler-Cope (1989). Insight without awareness. *Quarterly Journal of Experimental Psychology.* 41A (3), 553-577.

[27] R. Sun, (2002). *Duality of the Mind. A Bottom-Up Approach Toward Cognition.* Lawrence Erlbaum Associates, Mahwah, NJ.

[28] R. Sun and X. Zhang, (2002). Top-down versus bottom-up learning in skill acquisition. *Proceedings of the 24th Annual Conference of the Cognitive Science Society, Fairfax, VA.* Lawrence Erlbaum Associates, Mahwah, NJ.

[29] R. Sun, E. Merrill, and T. Peterson (2001). From implicit skills to explicit knowledge: a bottom-up model of skill learning. *Cognitive Science.* Vol.25, No.2, pp.203-244.

[30] R. Sun, (1999). Accounting for the Computational Basis of Consciousness: A Connectionist Approach. *Consciousness and Cognition* 8, 529-565. Also available online at http://www.idealibrary.com (article ID ccog.1999.0405)

[31] E. D. Taillard, L. M. Gambardella, M. Gendreau, J.Y. Potvin, (1998). Adaptive Memory Programming: A Unified View of Meta-Heuristics. *Technical report IDSIA-19-98*, IDSIA, Lugano, 1998. European Journal of Operational Research 135 (1), 2001, 1–16

[32] E. Tulving, D.L. Schacter, H. A. Stark, (1982). Priming effects in word-fragment completion are independent of recognition memory. *Journal of experimental psychology: learning, memory, and cognition*, 8(4), 336-342.

[33] E. K. Warrington, L. Weiskrantz (1970). Amnesia: consolidation or retrieval? *Nature*, 228, 628-630.

[34] C. Watkins, (1989). *Learning with Delayed Rewards.* Ph.D. thesis, Cambridge University, Cambridge, UK.

[35] D. Willingham, M. Nissen, and P. Bullemer, (1989). On the development of procedural knowledge. *Journal of Experimental Psychology: Learning, Memory, and Cognition.* 15, 1047-1060.

Index

Appendix A

Berry and Broadbent Experiment details

This appendix contains numeric results from the Berry and Broadbent 1989 experiments reported in [6], as the main text contains only graphical representation of these results. Table A.1 contains the experimental results, while Table A.2 lists experimental results.

Group	Set 1	Set 2	Set 3
salient control	5.83	6.92	10.42
salient experimental	6.75	12.25	14.83
non-salient control	4.17	5.83	6.83
non-salient experimental	4.33	2.50	11.75

Table A.1: Numeric results of the Berry and Broadbent experiments

Group	Set 1	Set 2	Set 3
salient control	8.54	11.51	12.60
salient experimental	8.50	11.80	16.82
non-salient control	4.74	4.92	4.88
non-salient experimental	4.69	4.38	10.61

Table A.2: Numeric results of the sumilation

Appendix B

Mathematical derivations and proofs

This appendix contains some mathematical derivations and calculations which were not included in the main body of the work, because in the author's judgement they did not bear direct relevance to the subject matter, or could be of interest

B.1 Effect of weight scaling on CLARION output

In this appendix, the effect of scaling the weight contributions of top and bottom levels is explored. In CLARION output formula 2.6 (page 18), it can be seen that the quantity affected by the weights is $Q'(a_i|x)/\alpha$. By equation 2.5 this quantity is:

$$Q'(a_i|x)/\alpha = (W_l * l_i + W_r * \sum_j r_j(a_i|x))/\alpha = (W_l * N + W_r * M)/\alpha \quad \text{(B.1)}$$

where N and M are recommendations of bottom and top level for a given action a_i. Now for each W, substitute w as follows: $w_l = \frac{W_l}{W_l+W_r}$, and $w_r = \frac{W_r}{W_l+W_r}$ which is equivalent to scaling condition 2.7. Substituting into the original equation for $Q'(a_i|x)/\alpha$:

$$(W_l * N + W_r * M)/\alpha = (w_l * N + w_r * M)/\frac{\alpha}{W_l + W_r} \quad \text{(B.2)}$$

Therefore for any pair of weights W_l, W_r in CLARION, it is possible to find such temperature α', such that probability for the model of choosing choosing every action $p(a|x)$ remains the same after W_l and W_r are scaled in relation to each other.

There are two significant consequences of this fact for this study:

1. Since throughout the experiments temperature, α, remained constant, scaling had the effect of increasing randomness in system output.

2. For complete parameter search either scaling has to be undone, or temperature should become a search parameter.

B.2 Calculating a scaled error function

A standard error function in the optimization problem was euclidean distance between the human and model results, given in equation 5.1 in Chapter 5. However, multiplication of the results by some constant a is allowed, then a much closer match on the simulated trends trends might be obtained. The constant a needs to be found before the error function could be computed. The problem is stated as follows. Given distance:

$$D = \sqrt{\sum_{i=1}^{12} (a \cdot x_i - y_i)^2}$$

find the value of a, where D is at its minimum. To solve this[1], note that partial derivative of D^2 with respect to a is:

$$\frac{\delta D^2}{\delta a} = 2a \sum_{i=1}^{12} x_i^2 - 2 \sum_{i=1}^{12} (x_i \cdot y_i)$$

D^2 is used here instead of D for simplicity of calculations. The square of the distance, D^2, will reach its minimum at the same point where D does. The derivative has the value of 0, and changes it sign for the following value of a:

$$a = \frac{\sum_{i=1}^{12} (x_i \cdot y_i)}{\sum_{i=1}^{12} x_i^2}$$

This is the value of a where the euclidean distance between scaled model and human results will be minimal.

[1] The problem can also be solved geometrically by noting that $a \cdot |X| = |Y| \cdot cos(X, Y)$.

Appendix C

Technical remarks about the simulations

C.1 Obtatining and running the source code

Source code for the Stanley et al. simulation is available at the following URL: *http://kmicic.midamerica.net/projects/clarion/StanleySim-src.zip* In case the above URL is not functioning you may try requesting the source code by emailing the author at *wiedzmin@yahoo.com*.

Running instructions:

```
% javac *.java
% java mainDC
```

The code was ultimately compiled under jdk 1.3, but it should compile and work under jdk 1.1 and 1.2. Things will be very slow without the JIT compiler. Under jdk 1.3 (with JIT) on a PII-450 the simulation takes about 30 seconds. Without JIT you can expect to take 15-20 minutes.

Source code for the Berry and Broadbent simulation is available at the following URL: *http://kmicic.midamerica.net/projects/clarion/BBSim-src.tgz* Running instructions:

```
% javac *.java
% java personsim
```

To alter the number of subjects in the experiment (to achieve more predictable asymptotic behavior), edit NUM_SUBJECTS in World.java.

Source code for the Berry and Broadbent simulation optimization is available at the following URL: *http://kmicic.midamerica.net/projects/clarion/BBOpt-src.tgz* Running instructions:

```
% javac *.java
% java personsim
```

Optimization is all done in SimpleScatteredSearch, and all that this class needs is a caller who has a function called computeError, returning a double. To alter the performance function, rename the desired function to computeError().

C.2 Computer system parameters

Most of the simulations and development were done in Borland's JBuilder Personal Edition (versions 7 and 8), using JDK 1.4, under Windows 2000. Initial development was done on the same machine, but under NetBSD in a text editor and run by JDK 1.1. The Just-In-Time java compiler, available in JDK 1.3 and higher, has speeded up the simulations by an order of magnitude.

The main machine where the simulations were run remained the same, but over the years it has undergone many makeovers. The running times are given for a Pentium II 450Mhz with 384Mb RAM.