Appeared in T.-W. Chan & J. Self (Eds.), *Emerging computer technologies in education: Selected papers of the International Conference on Computers and Education (Taiwan, December 1993),* pp. 207-42. Charlottesville, USA: AACE, 1995

Perceptual Recategorization vs. Grammatical Task Models —Relating Neomycin, Soar, and Neural Darwinism

WILLIAM J. CLANCEY
Institute for Research on Learning
66 Willow Place, Menlo Park, CA 94025 USA

Abstract: Educational applications of Artificial Intelligence have always been based on symbolic theories of memory, problem solving, and learning. Situated cognition research, as well as recent work in neuropsychology, has called these models into question. This paper attempts to bridge the two perspectives. Cognitive models based on an exclusively linguistic model of knowledge ("the symbolic approach") are related to Edelman's Theory of Neuronal Group Selection (TNGS), to examine how subgoal hierarchies are learned and problem spaces aligned with perceptual categorization. To make this argument accessible to a broad audience, I begin by carefully relating Neomycin to NL-Soar, explicating how a domain-general model constitutes a problem-solving grammar. A short tutorial on TNGS is followed by a discussion of how chunking, subgoaling during an impasse, and construction of new operators might arise from classification couples, global maps, and pre-linguistic conceptualization. The argument suggests how human learning gives rise to the patterns described in Neomycin and similar cognitive models. Finally, we briefly consider how this perspective reframes research on learning and instructional design.

Background: Situated Cognition And Intelligent Tutoring Systems

From the beginning of Intelligent Tutoring System (ITS) research, we based our design of instructional programs on contemporary knowledge representation frameworks (e.g., semantic networks, production rules, frames, blackboards). The idea that a body of representations of knowledge expressed in such a framework was knowledge itself drove nearly two decades of tutoring, student modeling, and explanation research (Wenger, 1987). In particular, instructional strategies were based on the idea of a difference model, by which variance between the student's abstracted behavior and the behavior suggested by the model was reduced to knowledge representations (i.e., facts, rules, problem states), and instructional mediation was based on these differences (though actual program actions might include hints, graphic explanations, or new problems).

In the past five years, the idea that knowledge bases are just models and not knowledge itself leads us to reconsider how we will use qualitative modeling techniques for instruction (Clancey, 1992b, 1993b; Sack, et al., 1991). The present paper begins a detailed examination of existing models and how they might be improved: Why do models like Neomycin and Soar work so well? What are their limitations for modeling human behavior? Can neuropsychological (neural network) approaches help us develop better symbolic models of student learning?

To address these questions, I provide a broad review of Neomycin and Soar, with an introduction to Edelman's Theory of Neuronal Group Selection (TNGS). By relating Neomycin to Soar, I hope to clarify the nature of symbolic models, particularly their stored (and bounded) grammatical view of knowledge. Through a discussion of TNGS, I show tentatively why human behavior appears grammatical (and hence why Soar and Neomycin work so well as descriptions of behavior), by explaining how subgoal hierarchies are learned. The idea I will develop is that conceptualizing new problem-solving operators occurs at a pre-linguistic level, with each operator coordinating perception of the problem being solved (the "working memory") and linguistic models (the "knowledge base"). In particular, I show how Neomycin's problem spaces are coupled to the focus of diagnostic tasks; so impasses in a problem space are resolved by reducing the focus grainsize of working memory (the patient, the situation-specific model, the differential, a hypothesis, a finding). I introduce a notation, called Interactive Control Structures (ICS), which links symbolic descriptions to neural processes. I use this notation to recharacterize chunking and hierarchical learning of operator sequences.

The connections made here are necessarily speculative, but I argue that such analysis and intermediate notation (a symbolic model acknowledged to be non-equivalent to human capability) are useful for progressing from today's cognitive models to a better understanding of human learning.

Current Understanding Of Neomycin

The Neomycin model of diagnosis (Clancey, 1988, 1992a) is a description of an experienced physician's problem solving. Tasks and metarules abstract patterns of behavior. Recurrent sequences (e.g., test hypothesis) are organized in a subgoal hierarchy. Rules for shifting attention and searching a general model of disease processes (the "knowledge base") construct an evolving model of the patient (the "situation-specific model").

Situated cognition research claims that the human expert is not literally executing the Neomycin model, in the manner of a computer interpreting a program. Rather, the model *describes how physician behavior appears*. We hypothesize that human capability is more improvised and open to change (Clancey, 1991b, c).

Diagnosing a patient is not merely instantiating procedures or applying rules rotely, but reconceiving and theorizing how terms relate to situations, what constitutes a problem, and how processes being comprehended causally relate. Diagnosis is a *story comprehension process*, in which the meaning of the patient's experience is more than a name of a disease and causal network. A diagnostic interpretation and understanding can be modeled by such linguistic descriptions, but the physician's *impression* of fittingness and similarity cannot be fully captured in (reduced to) words. Dewey's (1938a) study of inquiry and Schön's related work (1987, 1990) empirically illustrate and develop this idea.

Simply put, the physician is learning new ways of viewing patients and the diagnostic process, plus adjusting his way of talking about diseases in the very process of understanding any given patient. Beyond this, situated cognition research, including empirical evidence from other studies (Bamberger, 1991; Lave, 1988), suggests that what the physician perceives, that is, *what constitutes the data to be explained*, is also an adaptation, such that what he perceives, his understanding, and his action in the therapy setting constrain each other dialectically, as well as chronologically over time. Our interest in modeling the diagnostic process is to understand better where each new action comes from, in the form of a question about the patient, a reexamination of the patient or reconsideration of evidence, or a conjectured point of view.

Another central hypothesis is that story understanding ("inquiry" more broadly) involves transformations of previously constructed stories. We are especially interested in the conceptualization process which is pre-linguistic, by which new ways of seeing and talking arise, grounded in previous sensorimotor experience. We need to improve our understanding about how reifying objects, properties, and procedures in abstract terminology and relations (such as Neomycin's tasks and metarules) provides guidance to a student. In other words, given our rejection of the idea that knowledge is just stored words, definitions, and rules, we need to reconsider *how talk changes what people do*.

In more familiar terms, this investigation seeks to understand how "transfer" occurs—how does conceptualization of new situations arise from the activation and adaptation of previous ways of behaving? Again, we need to understand the transformations that are possible and which step outside of Neomycin's model: How is each diagnostic experience not just a reactivation or reconstruction of a previous *program*, but an adaptive modification of how patient features (symptoms) *appear*, what they *mean*, and how they relate in a *causal understanding* of disease processes? In terms of other existing models of learning (e.g., Iran-Nejad, 1987), we need to understand how self-regulation works within an existing level of conceptualization, and how impasses arise that force the person to jump to a new coordination.

Such epistemological shifts will only gradually work their way through our conception of instructional programs. For example, the idea that practice, what we do, is inherently based on pre-linguistic coordination (not a bedrock of propositions and linguistic models) suggests that errors cannot be eliminated from human performance. Rather, reflection and re-theorizing are inherent aspects of ongoing interactions with the indeterminate and open social, physical, and representational systems in which we behave. Teaching shouldn't be driven by a simplistic idea of driving out misconceptions and faulty procedures, leaving behind a body of guaranteed skills and models, but preparing students for a self-regulated process of modifying how they see and talk about things.

In laying out some basic relations between Neomycin, Soar, and neurological processes, this paper contributes to this re-investigation of the relation between human learning and computer models of knowledge.

Need For A New Modeling Framework

To better understand why current cognitive models work as well as they do, yet to open up an investigation of their limitations, we need some kind of intermediate modeling framework and language. For present purposes, we need a variation on Neomycin, which as a computational model will not be more powerful than Neomycin (it will not diagnose patients better), but will better fit the perceptual, conceptual, and coordination processes that we now believe underlie human performance and give rise to the behavior patterns we have described in Neomycin.

This intermediate description will be based on a "functional architecture" of how the brain works (again a linguistic description). This model will better fit the neurological processes of categorization and coordination that we believe are operating in the brain. The functional architecture we will use here is called "interactive control structures" (ICS); it is being used to redescribe a variety of human memory and learning phenomena (Clancey, 1991a). The original idea stems from Bartlett's notion that remembering is a form of recoordination, triggered by an impasse during a comprehension process, and composing a new perceptual detail within an ongoing story (Clancey, 1991b, 1992c, 1993a).

In effect, we are striving for an understanding of how perceptual categorization, conceptualization, coordination of habits and routines develop subconsciously in animals and people. We start by conjecturing a parsimonious compositional process by which these "levels" (perception, sequencing, conceptualizing, and speaking) are just *categorizations of categorizations* produced by a single neurological process of activation and subsumption. For the moment, we adopt the map-of-maps selectional learning process, described by Edelman and modeled in the Darwin robots, as a possible mechanism by which physical neurological structures are activated, sequenced, and composed (Edelman, 1992; Smoliar, 1989; Clancey, 1993a). This paper will not describe these neurological processes in detail, but focus instead on an intermediate level of description (interactive control structures, ICS). In effect, this paper is part of an ongoing theoretical effort to bridge neurological and descriptive models of learning. I begin by relating Neomycin to Soar because we need first of to better understand descriptive methods. The mapping of our exposition can therefore be summarized as:

Soar = Neomycin <—> Neomycin Prime (ICS) <—> Neural Darwinism

We view Neomycin and Soar as behavioral models, replicating how behavior appears to an observer over time. We view Edelman's (1987; Smoliar, 1989) Neural Darwinism as a neurophysiological model, replicating patterns of perceptual categorization and sensorimotor coordination. We develop here an intermediate model (named Neomycin Prime) which I characterize as neuropsychological because it will begin to show why the Soar and Neomycin descriptions work so well and how the patterns they describe could arise in neurophysiological processes.

In effect, we seek to reground ITS cognitive models in a model of human memory and learning. There is good empirical evidence for each level of description, but considerable analytic work is necessary to tie these descriptions together.

Quite possibly, we will conclude that this better understanding is of minimal value to ITS. But this in itself would be a major finding, given that most of the research of the 1970s and 80s assumed that better understanding human memory and learning was fundamental to better teaching and instructional design. We might conclude that existing, formal linguistic models of cognition are good enough and that we don't really care how conceptualization arises. But first cognitive science needs a neuropsychological mapping to be worked out.

The Problem Addressed Here: Relating Theory To Practice

Probably the most well-known model of human learning is the Soar system, "a symbolic architecture for intelligence that integrates basic mechanisms for problem solving, use of knowledge, learning, and perceptual-motor behavior" (Cho, et al., 1991; Laird et al,. 1990). Indeed, our first problem as scientists of human learning—before we can jump to a neurological level—is to understand how Soar relates to other models. In particular, how does Soar relate to models of human problem solving and learning commonly used in educational computing? In this paper, we consider in detail how Soar is related to Neomycin.

The most striking similarity between Soar and Neomycin—plus of course most models of human knowledge—is a subgoal hierarchy, which in Neomycin is called a *task hierarchy* (Figure 1). For example, the task hierarchy of Neomycin (including the metarules for diagnosis) distinguishes between "Group-and-differentiate" and "Explore-and-Refine." These terms model two processes we observe (and that a physician talks about): "looking up to broader categories of disease" vs. "looking down to elaborate and evaluate conjectured disease processes." This paper seeks to explain how such a hierarchy is learned by examining its structure in more detail. We build on the hypothesis that pre-linguistic, conceptualization processes enable people to learn procedures bottom up. By conjecture, this is the same process by which grammatical speaking is learned (e.g., see Edelman, 1992; Lakoff, 1987).

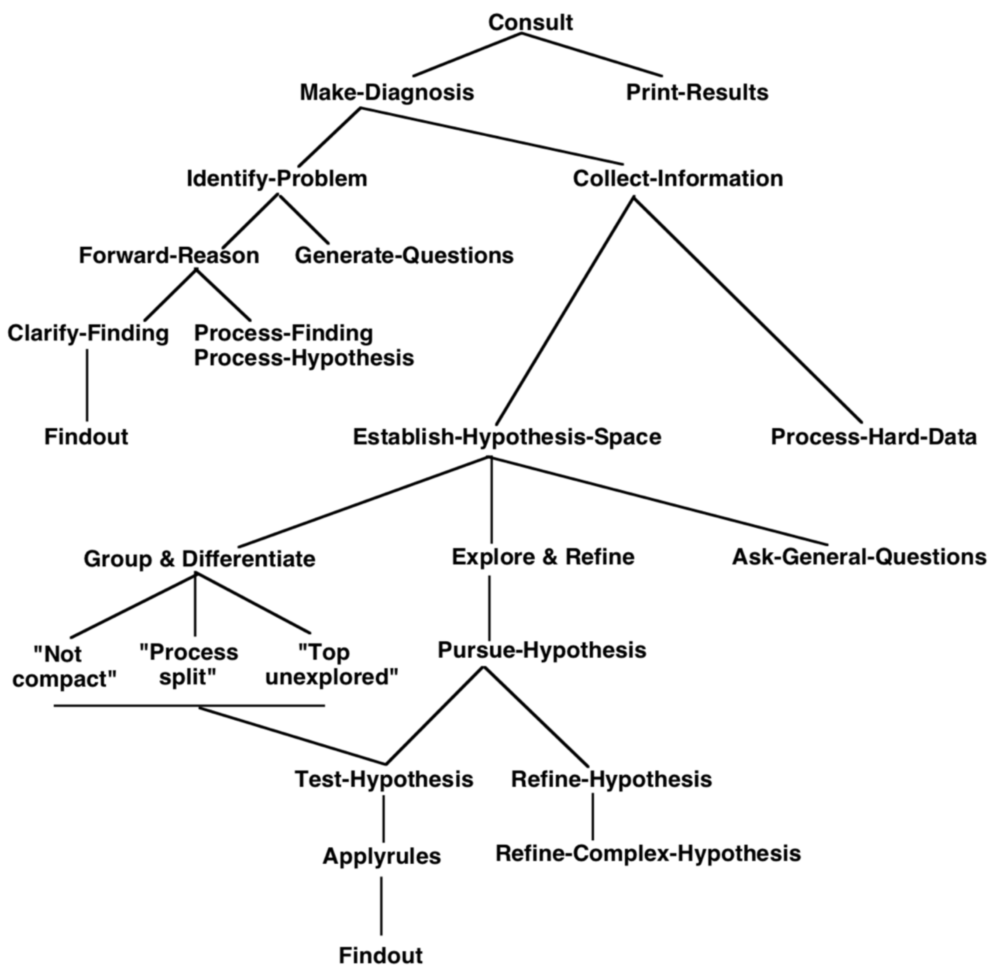

Figure 1. Neomycin hierarchy of diagnostic tasks

According to our situated cognition perspective, such hierarchies describe attention shifts we observe during human problem solving, not stored descriptions (schemas, rules, or programs) in the expert's brain[1]. The situated cognition view of memory claims that there is no inherent vocabulary of terms, relations, and metarules that human behavior will fit precisely (Rosenfield, 1988; Clancey, 1993a).

How do patterns of behavior, commonly called habits, develop before a person can articulate categories describing these patterns? In effect, we are pursuing Ryle's analysis of practice (what people can do) and theory (how people talk about what they do):

> Efficient practice precedes the theory of it; methodologies presuppose the application of the methods, of the critical investigation of which they are the products.
>
> It was because Aristotle found himself and others reasoning now intelligently and now stupidly and it was because Izaak Walton found himself and others angling sometimes effectively and sometimes ineffectively that both were able to give their pupils the maxims and prescriptions of their acts.
>
> It is therefore possible for people intelligently to perform some sorts of operations when they are not yet able to consider any propositions enjoining how they should be performed. (Ryle, 1949)

Hence, our first goal is to understand how people (and other animals) can conceptualize and perform procedures before they have a language for rationalizing what they do. In particular, we know that Neomycin's tasks and metarules go beyond what a typical physician can say about the diagnostic process, just as a natural language grammar goes beyond what a child can say about speaking.

Broadly put, the second goal is to understand how people are more flexible than Neomycin: How can people step outside a vocabulary, a disease model, a diagnostic procedure, as it appears on paper, to do something new? We know people can accomplish more than any such model describes, by the simple observation that no computer program can yet create such models on its own. There must be some other process (by conjecture involving more than the manipulation of descriptive models) by which people reconceptualize and recoordinate their behavior during problem solving itself. Fundamentally, this view argues that learning is a primary phenomenon, inherent in every perception and action, not just a reflective process occurring when trouble occurs or after a problem is solved. Practically speaking, such a comparison of Neomycin to human problem solving would suggest and help explain how descriptive models might fail to model students.

In summary, this paper provides a broad conceptualization for a new computer model of diagnostic reasoning, relating Soar, Neomycin, and evolving neurological models. Subsequent steps might involve protocol analysis (to show how this revised model better fits already collected data) and developing a computer simulation (e.g., indicating during the running of Neomycin where a human's performance might differ). These are all important, useful steps prior to building a program with human capability.

We will now proceed logically through several steps of exposition and analysis:

1. Review how Neomycin's tasks are *operators* for building a situation-specific model, which detect and fill gaps with respect to the general domain model (Clancey, 1992a).
2. Show how Neomycin relates to Soar (in particular, how the task hierarchy and metarules relate to problem spaces and operators).
3. Show how the grainsize of subgoals strictly corresponds to levels of abstraction of the situation-specific model (illustrated by comparing NL-Soar (Lehman, et al., 1991) to Neomycin).
4. Using the ICS representation of the theory of neuronal group selection, show how problem spaces are coupled to perceptual recategorization.
5. Using the ICS notation, show how higher-order goals are *conceptualizations* of operator sequences, where concepts are not equated with descriptions, but are physical coordinating processes.

---

[1]Although Neomycin was designed as a cognitive model, the emphasis is on hypothesis generation and testing, not interpersonal interaction. For example, "Print-Results" replaces the human expert's explanations to the patient; and no consideration is made for how information is gathered in practice from patients, colleagues, medical records, etc.

Besides showing how Neomycin is just a simplification of Soar, this analysis has the theoretical value of beginning to relate neural net models to complex problem solving models. Practically, the analysis suggests how instructional strategies might relate a student's conceptualization difficulties to perception and attention sequencing.

The Neomycin Problem Solving Model

In what is now called the "classic" symbolic approach of AI, the reasoning process is modeled by an inference procedure that applies a general model to facts and constraints in some problem-solving situation to derive a situation-specific model (SSM) (Figure 2). The SSM constitutes a plan, a design, a diagnosis, etc. or a sequence of these. The key step in creating Neomycin from Mycin was expressing the inference procedure in terms of variables (e.g., "finding," "hypothesis," "rule"), rather than domain terms. Further analysis showed that Mycin's SSM, called a context tree, is equivalent to what is called a "blackboard" in many expert systems[2].

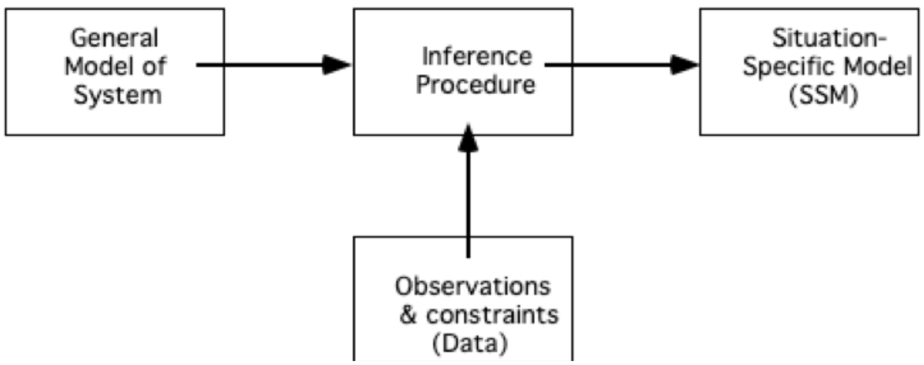

Figure 2. An inference procedure is a program for gathering information about system behavior and the environment (data) in order to make assertions about the system producing this behavior (the task called "diagnosis"), the system that could produce this behavior ("design" and "control"), or how a system will behave ("prediction").

During the growth period of expert systems research, roughly 1975-1985, a number of representational formalisms were invented. These formalisms adopt different perspectives on the nature of inference, summarized in Figure 3. Inference can be described in terms of operators for searching the domain (general) model, subgoaling when encountering impasses, or constructing a SSM. In each case, the overall reasoning process fits Figure 2, but researchers emphasized different search perspectives. I point this out because it explains why I didn't realize until recently the close relationship of Soar and Neomycin. This relationship becomes apparent when we study NL-Soar, but first we must review the nature of the SSM in Neomycin.

The Situation-Specific Model

Understanding the nature of a situation-specific model requires first understanding the structure of the general domain model. In Neomycin, the domain model consists of a classification of disease processes and a causal network relating symptoms to abnormal structures and processes in the body. Patient findings are explained in terms of the disease processes that cause them or that they might cause. Figure 4 shows a simplified version of Neomycin's domain model (which I have called the general model in Figure 2 and domain theory space in Figure 3). This figure is annotated to show one way of viewing two diagnostic subtasks: Suppose that a disease process has been hypothesized (e.g., a certain kind of headache of a long duration suggests chronic meningitis). The subtask Group-and-Differentiate "looks up" to consider and

[2]The material in this section, including all of the figures, is excerpted from "Model Construction Operators" (Clancey, 1992a).

contrast diseases categorically. The subtask Explore-and-Refine "looks down" to consider specific causal and subtype relations.

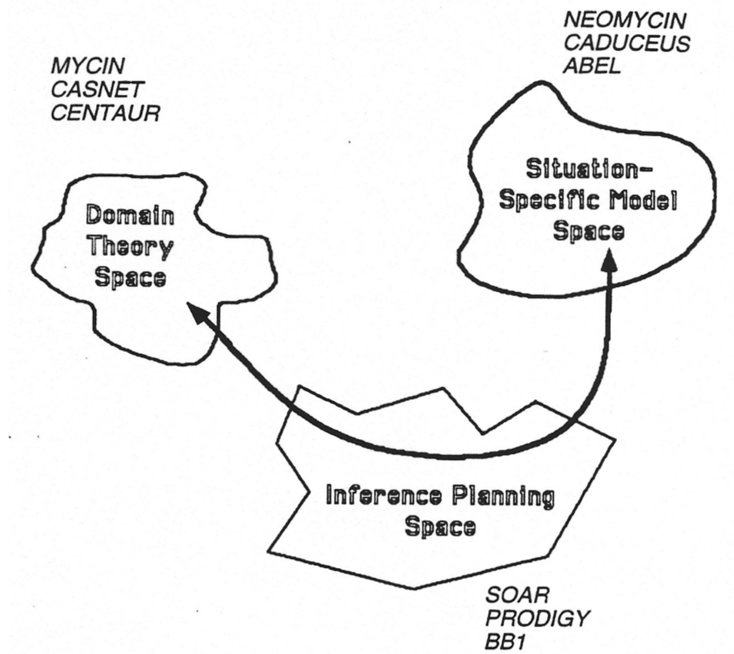

Figure 3. Alternative Spaces for Defining Inference Operators

This description (Figure 4) corresponds to the left side of Figure 3, the domain theory view of inference. In contrast, the SSM view considers how a given domain model is copied over and assembled in a situation-specific view of the world. In effect, the nodes and links in Figure 4 will be replicated in the SSM, based on what is believed to be happening in a given patient's life (see Figure 5).

An SSM (or blackboard) has the form of *a graph constrained to have a certain form.* The form, specified as constraints on the structure of the graph, arise because the graph is not an arbitrary network, but is a representation of processes occurring in a physical system, constructed for a purpose. That is, the graph is a model.

For example, consider the form this graph takes for medical diagnosis. According to the Neomycin general model, the description "Viral-Meningitis" need not be elaborated into causes of viral meningitis, because all subtypes have the same therapy (orange juice, aspirin, and rest). If our purpose were scientific description rather than medical therapy, we would want to know what causes Viral-meningitis. Hence, the adequacy of the model is with respect to the discriminations it must make between therapy alternatives. This is of course precisely the idea of heuristic classification, in which classification models are chained in a sequence of steps: Patient —> disease —> therapy.

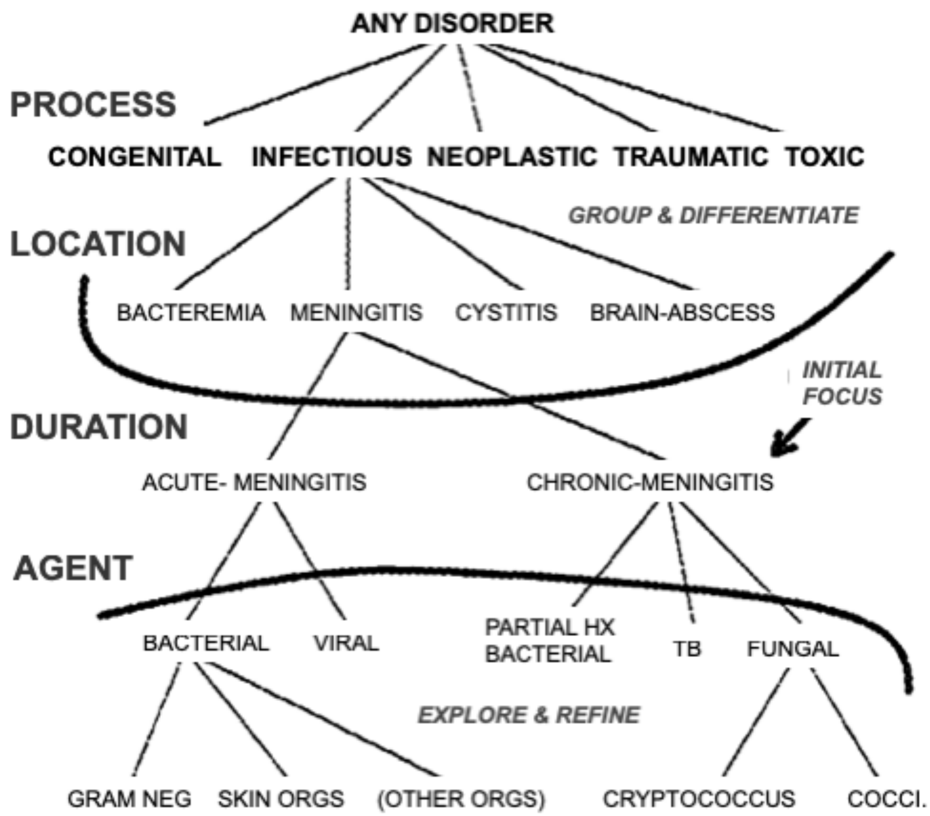

Figure 4. Looking up and looking down in diagnostic search: viewing inference as searching the domain model. (Redrawn from Clancey, 1992a, Figure 14, p. 23)

Neomycin's subtasks are operators that examine and modify links in the SSM (Figure 5). The key idea is that the SSM is inspected by the task operators during reasoning, such that the partial state of the SSM drives the inference process. The process of searching the general model (e.g., the disorder taxonomy) is secondary, a matter of finding the processes that might be occurring in this case, driven by gaps in the SSM.

Subtasks in Figure 5 are indicated by abbreviations. Roughly speaking, the reasoning process can be viewed as constructing a model of processes in the patient's body that are causing the patient's complaints. For example, information about a headache triggered rule 424, which caused a question about stiff-neck-on-flexion to be asked and meningitis to be hypothesized (subtasks Process-Finding abbreviated PF, and Applyrule! abbreviated AR!). The subtask Group-And-Differentiate (G&D) looked for support for meningitis by considering categorical evidence (for infection-process, a more general process description).

To use the node and link language more explicitly, Test-hypothesis grows links downwards, here placing febrile and a causal link (Rule423) below infectious-process. That is, TH finds causal support for a process description, and the support is indicated by findings attached below a node. Similarly, Refine-Hypothesis grows subtype links upwards, here placing acute-meningitis and then acute-bacterial-meningitis in the SSM. In this way, the final diagnosis (the most specific) is found at the top of the tree, with evidence hanging below it. An ideal diagnostic model is just one tree, containing all abnormal findings.

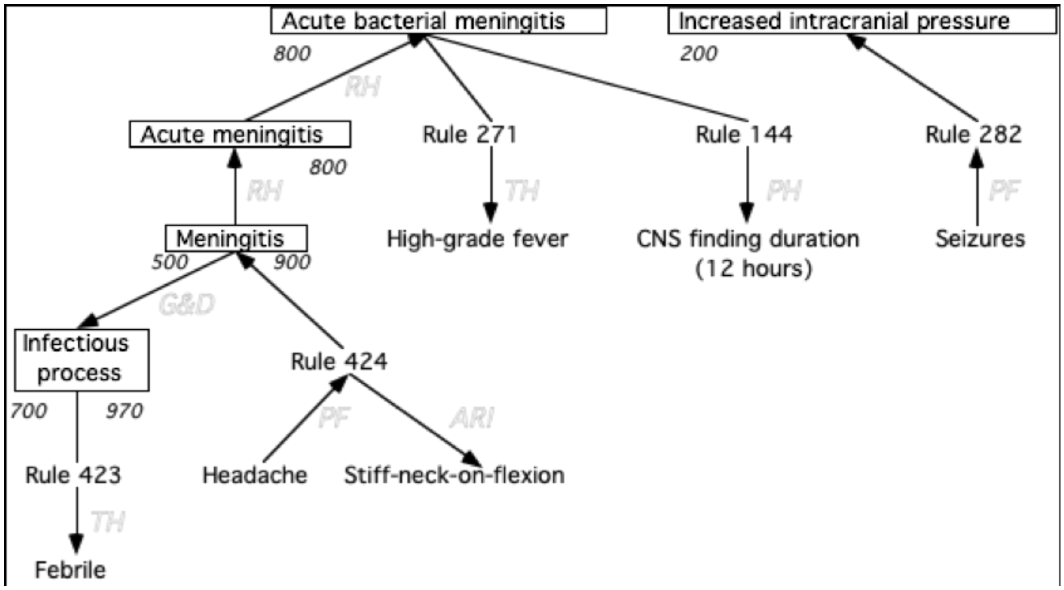

Figure 5. Diagnostic subtasks shown as operators for constructing a situation-specific model (arrows indicate the order in which nodes were linked; see text).

Examining Figure 5, notice that some of the subtasks mentioned in Figure 1, the complete task hierarchy, do not appear. The subtasks in Figure 5 are *primitive operators* because they directly place links in the SSM. More abstract operators control the order in which these primitive operators are applied. They are called *procedural operators* (Figure 6). The metarules for such subtasks constitute *ordered, conditional steps* in a procedure. In contrast, the metarules for primitive subtasks are *alternative methods* for accomplishing one thing—placing a certain kind of link in the SSM graph.

For example, a typical procedural operator is Pursue-Hypothesis. Its two metarules invoke Test-Hypothesis and Refine-Hypothesis in sequence. Procedural operators represent preference between primitive operators. To accomplish a top-down, breadth-first search of the disease taxonomy in Neomycin, we test a hypothesis before refining it. Analyzing the subtasks and metarules in this way revealed that Group and Differentiate could have been implemented as explicit subtasks in Neomycin, analogous to Test and Refine.[3]

In summary, Neomycin's subtasks can be formally described in terms of the particular nodes or links they place in the SSM, with more-abstract operators controlling how these primitive operators are applied.

[3]Metarules for other subtasks, such as Forward-Reason and Process-Hard-Data, not shown here, can be abstracted and regrouped in a similar way.

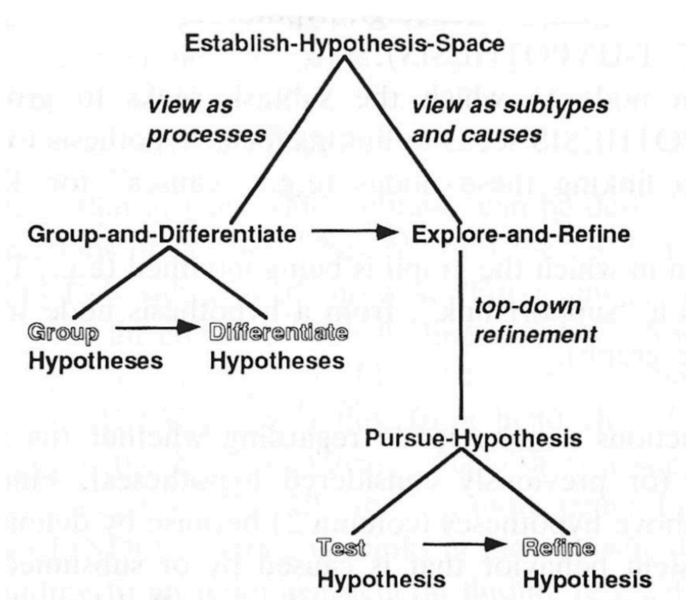

Figure 6. Procedural subtasks order the application of primitive operators (Group, Differentiate, Test, Refine) or other procedural subtasks (G-and-D, E-and-R). Arrows indicate preference relations represented by the order of metarules for a given subtask.

Learning New Relations And Subtasks

To understand how the inference procedure (the subtasks) is learned by people, we must understand how the structure of the *metarule language* (the relations of metarule preconditions) relates to the structure of the general model and how the structure of the *task hierarchy* (the foci of tasks) relates to the structure of the SSM, shown schematically:

Subtask focus <—> SSM
|
General Model <—> **Metarule Relation**

In the development of Neomycin, we realized the first part of this insight (how the general domain model is related to metarule premise relations), but didn't grasp the second (how the SSM is related to subtask focus).

We observed in the development of Neomycin that writing a new metarule almost always required defining a new domain relation, which further subclassifies the distinctions made before. For example, CAUSES becomes ENABLING-CAUSE and CIRCUMSTANTIAL-CAUSE. Thus, each new relation effectively creates a subset, intersection, or composition of the domain terms or rules defined by previously existing relations. The subtasks can therefore be viewed as operating on sets of findings, hypotheses, and domain rules, collecting, sorting, and filtering them in order to control how subtasks are accomplished. Relations in metarule premises (e.g., SUBSUMES, ENABLING-CAUSE) serve as conditions by which domain terms and rules are retrieved from the knowledge base, collected, and passed on to subtasks. We therefore find it convenient to state the metarules in terms of set operations, for example, "select *the set of findings* that are enabling causes of the hypothesis; select *the set of rules* that link these findings to the hypothesis; then apply these rules." Some of the different views adopted by subtasks are represented in Figure 6.

More generally, *different interpretation procedures require different views of the model (or process) being reasoned about*. These views take the form of new relations by which elements of the general model are classified, hence new structure is added to the model. In Neomycin, there are two process models structured in this way: The model of disease processes (the general model) and the model of the diagnostic process (the tasks and metarules). The metarules reason about the domain model. The compiler, explanation, and student modeling programs reason about the inference procedure.

For example, domain relations appearing in Neomycin's metarules are classified so the compiler can replace them by Lisp code. Other properties of domain relations are used by the explanation program (e.g., if one relation implies another, it may be sufficient to state just the more specific clause). The subtasks are classified in one way for Guidon-Manage's hint generator and classified another way for the explanation and student modeling programs (e.g., Guidon-Manage doesn't suggest subtasks below the level of subtasks in the student's menu). Image (London and Clancey, 1982) must know how metarules can be reordered or omitted when generating advice from a student model. In essence, these *relations act as filters* by which different elements of the (general) process model are preferentially collected and sorted for incorporation in the SSM being constructed (which may be a diagnosis, a compiled program, an explanation, a student model, etc.).

Domain-Generality And Grammars

As has been reported elsewhere, the Neomycin model was originally invented in order to facilitate explanation and student modeling in an instructional program. The resulting method directly parallels the approach followed in models of natural language generation and recognition. Neomycin’s metarules constitute a grammar for the diagnostic process. This was accomplished by using variables, rather than domain terms in the metarules (analogous to using variables like "noun" and "verb" in natural language grammars, rather than specific words). The relations serve to classify the findings and hypotheses, in the same manner that a natural language lexicon classifies words (e.g., passive verb, demonstrative pronoun), determining which grammar rules will control their assembly into sentences. The subtask structure serves as a higher-order representation of the entire consultative interview, analogous to a discourse grammar.

This observation becomes central in understanding the relation between NL-Soar and Neomycin. At another level, I am suggesting that learning to diagnose multiple patients is like learning to speak in different situations. A certain process of abstraction must occur by which general procedures are learned for applying the domain model and constructing the SSM. In terms of natural language, this procedural capability enables us to find appropriate words (from the lexicon, the general model) at the same time we construct a coherent story (the SSM), fitting both the form and topic of the conversation.

To understand how such grammars are learned—which after all was our assumption in using Neomycin for instruction—we consider how Neomycin is related to symbolic models of learning. Then we can consider how these models might be improved to better fit human performance.

Relating Neomycin To Soar

Developing a mapping between Soar and Neomycin requires understanding how the term “impasse” is used in Soar and the nature of chunking. We begin with some basic terminology: A Neomycin subtask is called a goal in Soar; Neomycin’s metarules correspond to operators for accomplishing a goal. Soar’s working memory corresponds to the SSM (called a “situation model” in NL-Soar). Just as metarules for procedural subtasks determine which primitive subtasks to apply, productions in Soar’s long-term memory can determine preference for competing operators to apply.

Unlike Neomycin, Soar’s productions are fired in parallel, so any production that matches elements in working memory will fire, a process called recognition. When Soar cannot proceed by recognition alone, an impasse occurs. In effect, this is the process of invoking a subtask to carry out a higher task (e.g., invoking Test-Hypothesis to carry out Pursue-Hypothesis)—that is, moving into another problem space. In Soar this explicit invocation may not be necessary, due to the result of prior chunking. To implement chunking in Neomycin, each time the program reaches a primitive task, which places an element in the SSM, we would bundle together the line of reasoning through the subtasks to create a new production rule. This rule would necessarily compose the general model and the diagnostic procedure into a domain-specific rule. The rule would be placed into long-term memory. Subsequently, when encountering a similar situation, Neomycin could go directly to this low-level operation in a single step (assuming that the precondition matching occurs in parallel).

Hence, the fundamental difference between Neomycin and Soar is the idea of chunking. Neomycin maintains a separation of domain model and diagnostic procedure. Soar continuously composes them, saving the path and variable bindings in a new rule.

Each level in the task hierarchy of Neomycin corresponds to a problem space in Soar. As in Neomycin, the operators in a problem space are defined by the person who wrote the program; the operators construct an SSM (working memory) and are invoked on the basis of what is in the SSM at any given moment.

Understanding Neomycin And Soar Terminology

The summary I have given here eluded me for more than a decade, despite the development of Neomycin-Soar (Washington and Rosenbloom, 1988), which provides a mapping from Neomycin to Soar. Part of the problem is terminological, as can be seen in this summary of Neomycin-Soar:

> Neomycin-Soar has a top-level problem space that reflects the basic generation and selection processes in classification. Possible hypotheses are represented as operators in this space. Generation consists of proposing operators, and selection is deciding among the competing operators. The hierarchical subprocedures in Neomycin that generate hypotheses are mapped into a hierarchical set of problem spaces for hypothesis-generation in Neomycin-Soar—the basic steps of the subprocedures map onto more problem-space operators, and the subprocedure control maps onto additional operator selection knowledge (encoded as productions). The selection subspaces are not implemented, but would use many of the same subprocedures. (p. 6-7).

The first terminological hurdle is the use of the term "operator" in Neomycin-Soar to refer to *a disease hypothesis*. This mapping exists nowhere else in the medical AI literature; in medical AI, "operator" is used to refer exclusively to diagnostic subtasks, such as Group and Differentiate. But in Soar search is treated uniformly—both the domain space and the diagnostic procedure space are represented as a hierarchy of operators. Specifically, *the production rules themselves*—both domain rules and diagnostic metarules—are called operators. In Neomycin, we labeled the subtasks as operators and viewed the metarules as production rules that implemented them. In effect, we labeled the nodes of the diagnostic strategy (Figure 1) as operators; Soar researchers labeled *the links below each node* as operators. Table 1 summarizes these two perspectives[4].

The essential aspect of Neomycin-Soar's design, from the perspective of the present paper, is the mapping from Neomycin's *hierarchy of subtasks* onto a Soar *hierarchy of problem spaces*. This mapping again eluded me because I thought of search occurring only in the domain space, viewing the diagnostic taxonomy as a hierarchy of problem spaces. Indeed, the task Establish-Hypothesis-Space was originally named "establish-problem-space," following Newell's terminology (Newell and Simon, 1972). Rather than viewing both the domain and subtask hierarchies as being a hierarchy of problem spaces, I viewed the problem space as being on the domain side, consisting of the entire disease taxonomy, and viewed the operators as being on the diagnostic procedure side. This is how the terms were generally used in medical AI research (for example, see Pople, 1982).

By separating Soar terminology in this way, I failed to realize that Soar operators hierarchically relate problem spaces. When Washington and Rosenbloom say, "the basic steps of the subprocedures map onto more problem-space operators," they mean that the metarules (the steps) of subtasks (subprocedures) are problem-space operators. Furthermore, "the subprocedure control maps onto additional operator selection knowledge (encoded as productions)" means that the premises of metarules are used for selecting which operator at a given level (i.e., which metarule) to apply. In effect, we have the mapping shown in Figure 7.

[4]This mismatch of terminology and visualization of representational frameworks is typical of the cross-talk that has inhibited AI research in the past few decades. See (Clancey, 1993c) for additional examples.

Table 1. Relation of Neomycin and Soar terminology

Neomycin Terminology	Soar Terminology
domain hypothesis	domain goal
domain rule	domain operator
premise and action of the rule, plus properties of domain rule and domain findings indicating whether the rule is an antecedent rule, trigger rule, historical vs. laboratory data, certainty factor.	“productions implementing the operator”
a hierarchical level in the disease taxonomy	domain problem space
diagnostic subtask	diagnostic goal
metarule	diagnostic operator
premise of the metarule	“production implementing the operator”
subtasks: group-and-differentiate, establish-hypothesis-space, pursue-hypothesis, refine-hypothesis, find-out	generation problem spaces
subtasks: test-hypothesis	selection problem spaces (not implemented in Neomycin-Soar)

Figure 7 represents the mapping given in Table 1, but emphasizes that a problem space consists of a goal and operators. When recognition (direct matching of the SSM to productions in long-term memory) doesn't yield an action to follow (a domain rule to apply, assertion to add to the SSM, or finding to request from the user), an impasse occurs, moving to a new subtask (subgoal) and hence a lower problem-space of metarules (operators).

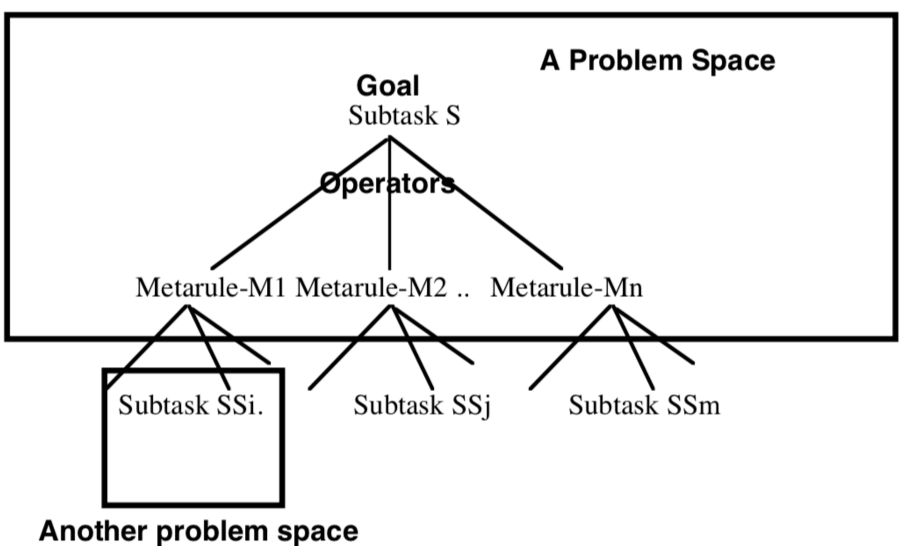

Figure 7. Relation of Soar terminology to Neomycin.

Relating Neomycin To NL-Soar

The mapping I have given here only became evident after I read a report on NL-Soar (Lehman, et al., 1991). I want the reader to fully appreciate the strong relation between Neomycin and Soar (which I believe is not apparent in the description of Neomycin-Soar alone) so I will present NL-Soar here.

The NL-Soar project "worked toward the goal of creating a general natural language capability for Soar" (p. 1). The underlying idea is that of a comprehension operator, which would generally make sense of the environment by bringing to bear information about the context, including ongoing problem solving, and general knowledge about the world. An interesting constraint imposed by the Soar architecture is that knowledge sources at all levels (lexical, syntactic, semantic, and pragmatic) should, through chunking, be accessed in a single recognition step. That is, there should be full integration of different processing perspectives, through practice.

An important addition to the Soar architecture, used by NL-Soar, is the use of "annotated models" produced by the comprehension process (Lehman, et al., 1991, p. 2). In NL-Soar, the annotated models include the model of the utterance (a parse structure) and the model of the situation in the world (including current and desired states). Annotated models are quite simply the situation-specific models I have described. Effectively, in using the term "annotated model" Soar researchers are designating part of working memory data structures as being coherent substructures that are chained together in problem-solving, representing states of the world, plans, and the meaning of representations (e.g., the utterance model). This chaining of models graphically shows that Soar programs accomplish the same model manipulation as Neomycin or blackboard programs in general. But also, the fact that NL-Soar is a model of natural language comprehension helps clarify the claim that Neomycin's diagnostic reasoning and Odysseus' student modeling processes (Wilkins, Clancey, and Buchanan, 1988) are formally equivalent to the parsing process of language comprehension.

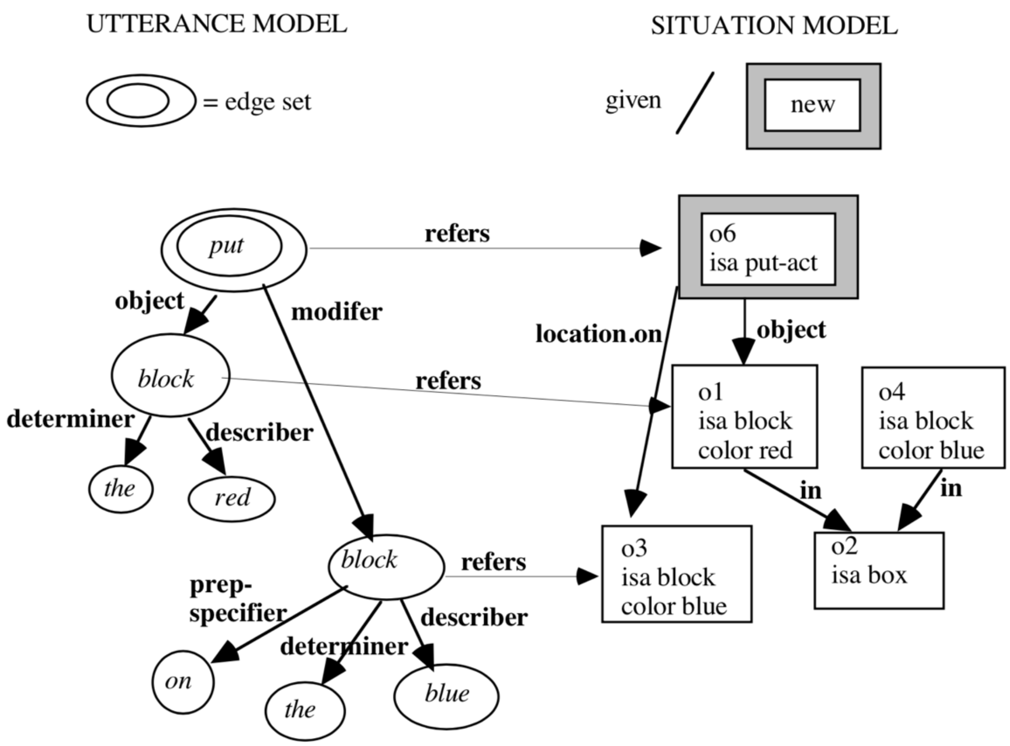

Figure 8. NL-Soar's situation-specific models of utterance and world context [simplified from (Lehman, et al., 1991; Figure 2-5)], corresponding to comprehension of the sentence, "*Put the red block on the blue block*."

Simply put, the NL-Soar model *views comprehension as parsing*, disambiguated by domain knowledge sources (semantics). There is a strict one-to-one mapping between input words and nodes in the annotated parse (structure of the utterance); this parse is mapped by *reference links* to the situation model (meaning of the utterance in terms of the present context) (see Figure 8).

The mapping here is similar to what occurs in explanation or instructional programs, in which the user makes an inquiry or request, which is mapped onto the program's model of the problem being discussed. For example, this is the mapping that occurs in Mycin's explanation program, when the user asks questions like, "Did you consider the patient's fever when concluding about the meningitis?" In this case, an utterance model must be related to both the model of the patient ("situation model" in NL-Soar) and the representation of the diagnostic process carried out by the program. This chaining of process of descriptions (domain <—> reasoning <—> explanation) is characteristic of many programs, but only recently have researchers acknowledged that the programs are *chaining representations of processes*. For example, Lehman, et al. note that the "situation" model data structure was only made explicit in Soar in 1988.

The presentation of NL-Soar's problem spaces in the NL-Soar report clarifies the basic relation between Neomycin and Soar (Figure 9).

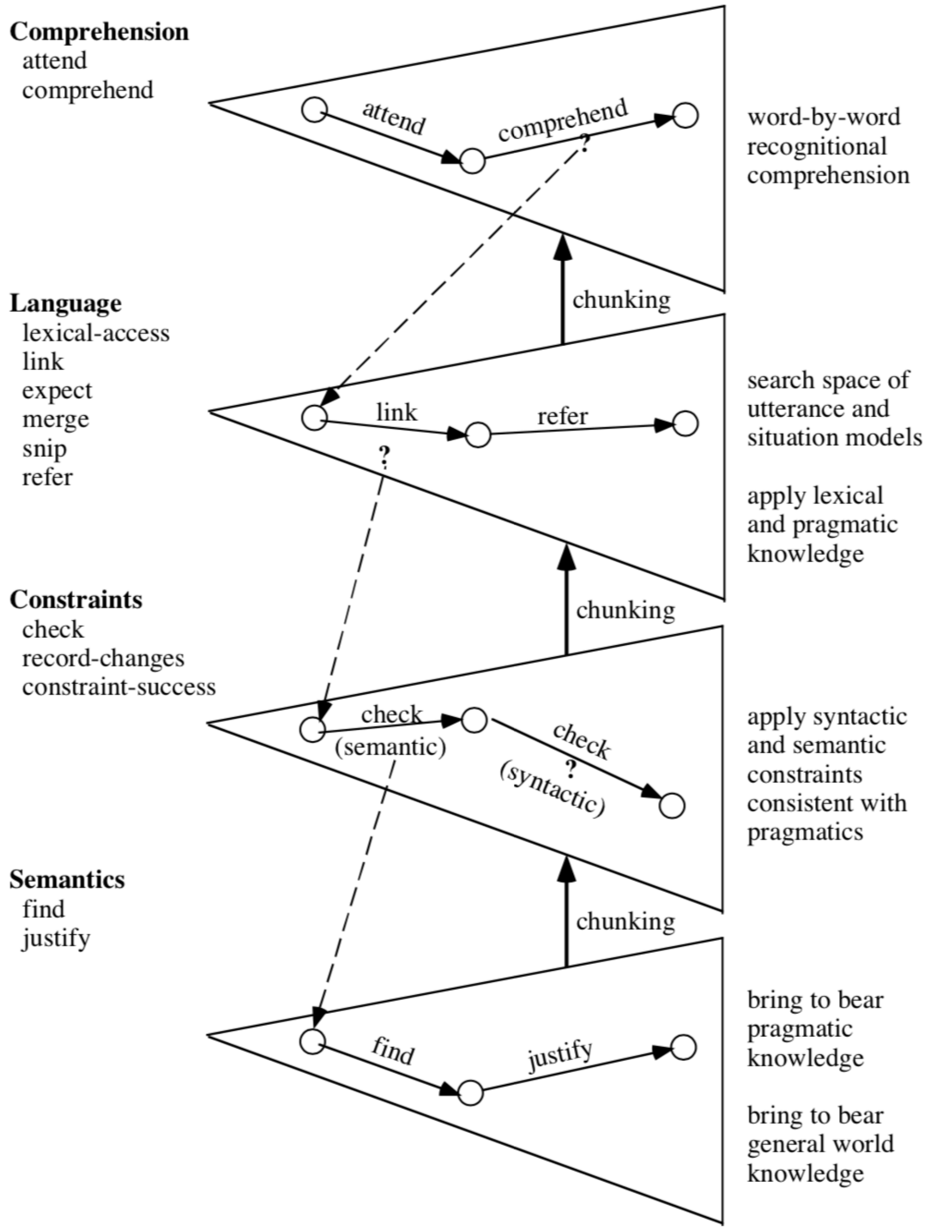

Figure 9. NL-Soar's problem spaces and operators (reproduced from (Lehman, et al., 1991; Figure 2-7))

The problem space levels correspond to the familiar levels of natural language processing: Comprehension, Language (utterance & situation models), Constraints (semantics, syntax), and Semantics (general world model, which they call "knowledge"). (Syntactic problems can cause impasses in the Constraints problem space, but there is no lower problem space for resolving them.) Again, if productions exist at a given level for accomplishing a goal, then no subgoaling (movement into a lower problem space) is necessary. After subgoaling occurs, a new production is composed (with variable bindings generalized) in the chunking process. This effectively moves syntactic, semantic, and pragmatic aspects of the model into the higher levels. As a result of this integration of diverse perspectives into new production rules, in some cases comprehension can proceed at the top level in one "decision cycle" without subgoaling: "chunking results in increased efficiency in each problem space in the hierarchy by transforming search in lower problem spaces into direct operator implementation in the higher space under similar circumstances" (Lehman, et al., 1991, p. 14).

To carry the analysis a step further, we can reformulate Neomycin's subtask hierarchy in this language of problem spaces and operators. Figure 9 makes clear that we should attempt to partition the subtask hierarchy into problem space levels (names on the left side of figure 9), and explicate the different aspects of the general (domain) model and situation-specific model that are brought to bear at each level. Also, we will show groups of metarules (subtasks) as labeled operators, connecting states (shown as circles in Figure 9).

In this adaptation (shown in Figure 10), I have simply mapped Figure 1, Neomycin's subtask hierarchy into the problem space representation of NL-Soar, working bottom up from the subtask Pursue-Hypothesis. To force the analogy between diagnostic story understanding and natural language comprehension, I have kept or only slightly modified the terminology in the figure. The mapping is striking, showing either that NL-Soar's structure is either quite general (fitting the diagnostic process well) or it is open to many interpretations. The key aspects of the mapping are:

- "word" is replaced by "finding"
- "utterance" is replaced by "findings"
- "lexical" is replaced by "domain"
- "syntactic and semantic" is replaced by "subtype and causal"
- "pragmatic knowledge" corresponds to the situation-specific model
- "general world knowledge" corresponds to the general domain model

There is of course no chunking in Neomycin, but the form of the diagram of NL-Soar is maintained to show how chunking could be implemented to allow recognition at higher levels and hence avoid invocation of lower tasks (subgoaling).

Ordering of metarules is shown as an ordering of operators (and hence states) within each problem space. For example, Establish-Hypothesis-Space is implemented as a set of ordered metarules, corresponding to Group-and-Differentiate, Explore-and-Refine, and Process-Hard-Data. These are states in the process of constructing a well-formed SSM (called the "differential" in medical diagnosis, when referring to the list of top-level, most-specific disease hypotheses).

A curious and important result of this analysis is shown by the parenthetical terms on the left side. Each problem space level is concerned with a different grain-size in the diagnostic story comprehension process, corresponding to the patient, the SSM, the differential (top level of the SSM), a hypothesis, and a finding. Moving downwards, *an impasses is resolved by adopting a new focus that is within and more specific than the focus of the current problem space*, as follows:

> We start by characterizing the patient in general terms (age, sex, etc.) and the originating complaint. The result is an initial set of hypothesis, represented in the SSM. Focus now shifts to the form of the SSM, gathering more information if there are no initial hypotheses (i.e., Generate-Questions), constructing a disease model based on history and physical signs, and ending by gathering laboratory information to support or refute the disease model. The third hypothesis space examines relations between the most specific hypotheses in the SSM (i.e., the differential), selecting hypotheses to be compared, generalized, or refined. An attempt is made at the end to throw new hypotheses into the pot by asking general questions that haven't been considered so far.[5] The fourth problem space focuses on hypotheses, which are supported or discarded by comparing the general domain model and patient information. Finally, the lowest problem space (not detailed here) focuses on findings, making inferences on the basis of subtype and definitional information.

---

[5]In particular, E&R is implemented as a set of preference rules for selecting a hypothesis to focus upon. The subtask invoked by these metarules is always Pursue-Hypothesis, which is omitted in Figure 10, because it is simply a shorthand name for two subtasks (Test and Refine).

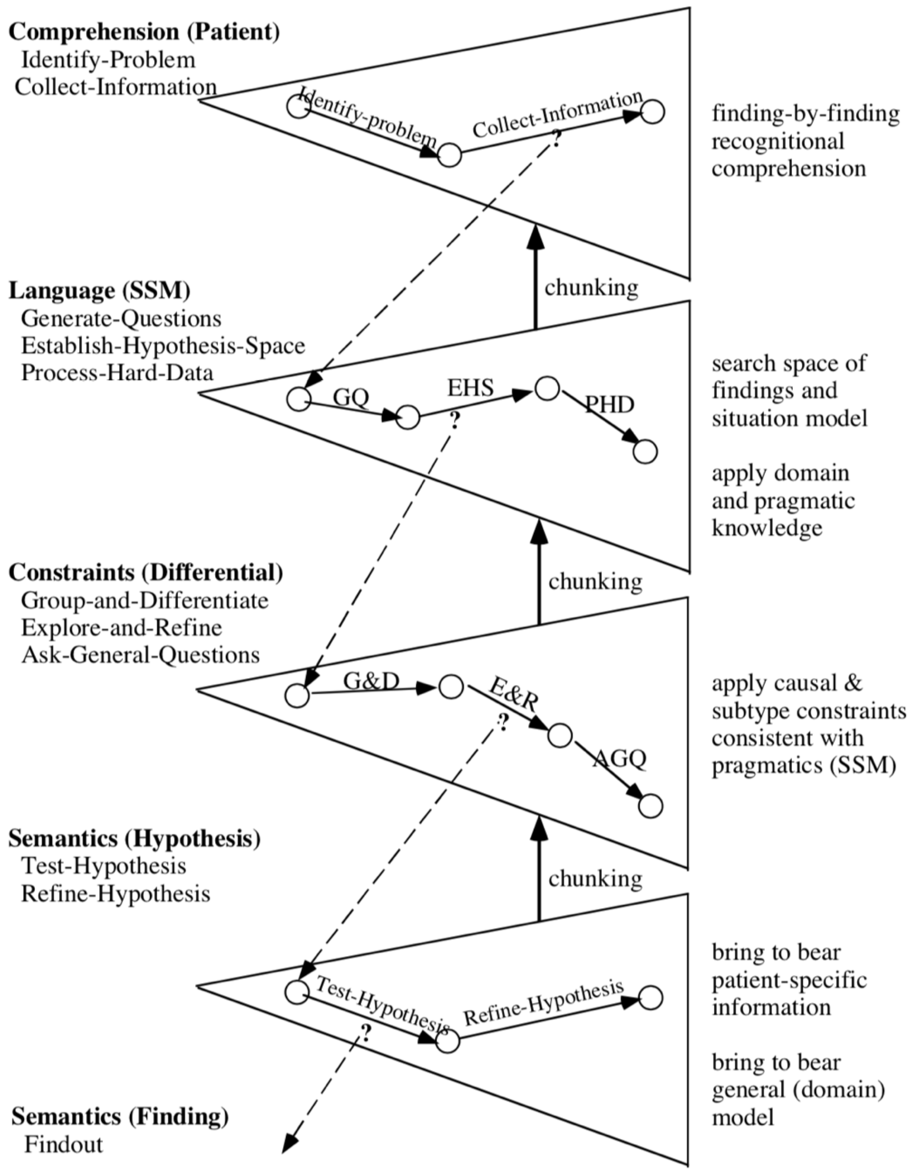

Figure 10. Neomycin's subtasks shown as problem spaces and operators

The story is more complicated, but not much. The operator Identify-problem is itself concerned with relating findings in the initial patient information (analogous to the utterance being parsed by NL-Soar). Hence, the same constraints brought to bear when comparing or testing hypotheses is considered directly here, but in the opposite direction, proceeding from findings to hypotheses, rather than hypotheses to findings. This subtask, not shown in Figure 10, which we call Forward-Reason, is invoked whenever new findings are entered into the SSM. In effect, this subtask corresponds to the immediate matching that always occurs on every cycle in Soar, as new elements in the working memory (SSM) are related to productions in long-term memory. That is, we don't need a representation for Forward-Reason in Figure 10; although it is implemented as an operator in Neomycin (for purposes of explanation to the user), such processing is part of the Soar architecture. The same observation holds for the subtask Apply-Rule.

The subtask Findout is represented in Figure 10 as a pointer to another problem space. Findout is actually invoked by operators at every level (Clarify-Finding within Identify-Problem; the subtasks

Generate-Question and Ask-General-Question; the Differentiate part of Group-and-Differentiate; and of course Test-Hypothesis).

From the perspective of learning, the broad pattern by which *each problem space focuses on a finer aspect of the story being constructed* is quite striking. In effect, this structure leads us to conjecture that the process by which operators are learned directly parallels a pseudo-perceptual process of envisioning the story at different grain-sizes: the patient, the SSM, the differential, a hypothesis, and a finding. That is, impasse resolution involves moving to a lower grainsize of analysis; chunking involves packaging this analysis into a single production at a broader context. Each problem space considers the story being constructed in terms of internal structure, involving components and relations not occurring a level higher. For example, to resolve an impasse in EHS, we examine the SSM to determine whether more general hypotheses have been considered (i.e., the "Group" part of Group-and-Differentiate). To resolve an impasse in E&R, we examine the most specific hypotheses in the SSM (called the differential) to select a single hypothesis to pursue (i.e., to test and refine). To resolve an impasse in Test-Hypothesis, we select a single finding to pursue (i.e., to rule out, generalize, or infer by definition).

After making some general observations about the nature of Neomycin and grammatical models, we will be ready to consider learning in more detail.

Grammatical Models

The direct mapping of Neomycin to NL-Soar illustrates why I call Neomycin a "grammatical model." A domain-general model of diagnosis is a model of comprehension, but at the level of constructing a causal story about the patient. The input utterances are facts about the patient, either provided or elicited. The SSM is a story of processes occurring in the body. The diagnostic problem is to create a story that fits the available information to the general disease models, at a level of detail useful for prescribing therapy. As indicated above, the domain model corresponds to a lexicon or vocabulary. The diagnostic procedure corresponds to a story construction process.

The student modeling programs built from Neomycin, called IMAGE and ODYSSEUS, are even more similar to NL-Soar. These programs take a sequence of student utterances (requests for patient information and stated hypotheses) and parse them in terms of the operators for constructing a coherent SSM. In effect, the operators of Neomycin correspond to the grammatical form of a diagnostic dialogue. As in NL-Soar, student modeling involves a mapping from text (student utterances) to a model of problem-solving processes (Neomycin's operators) to an SSM (model of pathological processes). That is, these student modeling programs are inferring the student's model of the patient in order to infer the diagnostic operators being used by the student. In effect, the modeling programs are trying to parse (understand) the actions of an agent who is trying to parse (understand) what is happening inside a patient.

We might show the three process models graphically as:

```
SPEAKING PROCESS —(about)—>
                DIAGNOSING PROCESS —(about)—>
                        PATHOPHYSIOLOGICAL PROCESS.
```

The chained parses (process models) refer to each other, as shown by the "refer" link in the description of NL-Soar (Figure 8). But between the UTTERANCE and SITUATION models of NL-Soar, we have interposed the PROBLEM-SPACE model of the agent. Put another way, for the purpose of student modeling, we split NL-Soar's SITUATION model into the models of the student's reasoning and the student's view of the patient[6].

Analysis relating expert systems, student modeling, and natural language comprehension is important if we are to realize the common principles by which diverse AI programs work. As is amply demonstrated by our reformulation of Neomycin, many researchers may not realize that their architectures are equivalent or how they might be simply mapped onto one another.

---

[6]The parsing process is more complicated because the student modeling programs maintain both an idealized expert's view (i.e., what Neomycin would believe and do) and a model of the student's view.

The description of NL-Soar establishes the principles by which grammatical models of comprehension (including diagnostic and student modeling programs) work. The components include:

- a domain model (including a lexicon or vocabulary) describes general objects and processes, organized by features, subtype and causal relations (e.g., general domain model of Neomycin);
- a syntax for structuring utterances (e.g., syntactic constraints in NL-Soar);
- a problem-solving model in terms of the agent's problem spaces and operators (e.g., subtasks and metarules of Neomycin);
- a parsing process for mapping between utterances to agent actions and belief model (SSM) (e.g., NL-Soar plus the procedures of Odysseus).

Relating Neomycin and its student modeling programs to NL-Soar constitutes a more complete, clearer description of the grammatical model of the comprehension process. Specifically, the analysis shows that in our student modeling programs, we have presumed that the student is making sense of the pathophysiological processes in the same way any adult makes sense of natural language utterances by mapping utterances to operations on the SSM. That is, we have modeled medical diagnosis in the same way that NL-Soar models natural language comprehension.

We are now ready to consider the problem of learning in more detail: What does this grammatical (stored lexicon and stored operator) view of comprehension leave out about human capability? How is such knowledge learned?

From Soar To Neural Models: The Nature Of Concepts

By the formulation of comprehension in NL-Soar, there is seamless representation and structuring of patterns and operators from the linguistic utterance level to the domain representations. At every level there are productions and feature mapping, chunking, and search. How does this bottom out? The researchers acknowledge that perhaps the most fundamental learning problem remains to be addressed. In the final paragraph of their report, Lehman, et al. (1991) say, "We have not solved the genesis problem (*Where does the knowledge in the lower problem spaces come from*?)..." (p. 39; emphasis in original).

By the hypotheses of situation cognition, there are non-linguistic elements and processing at the bottom levels. That is, below the level of domain terms, features, and relations, NL-Soar requires a pre-linguistic conceptualization space, akin to conceptualization and perceptual categorization in Darwin. Edelman's summary of these ideas is useful:

> The word "concept" is generally used in connection with language, and is used in contexts in which one may talk of truth or falsehood. I have used the word concept, however, to refer to a capability that appears in evolution prior to the acquisition of linguistic primitives....
>
> This recognition must be relational: it must be able to connect one perceptual categorization to another, apparently unrelated one, even in the absence of the stimuli that triggered those categorizations. The relations that are captured must allow responses to general properties— "object," "up-down," "inside," and so on. Unlike elements of speech, however, concepts are *not* conventional or arbitrary, do *not* require linkage to a speech community to develop, and do *not* depend on sequential presentation. Conceptual capabilities develop in evolution well before speech. Although they depend on perception and memory, they are *constructed* by the brain from elements that contribute to both these functions. (Edelman, 1992, p. 108)

Conceptualization, as hypothesized by Edelman's model, involves a composition of categorizations. That is, the process by which perceptual categorization occurs will occur again at a higher level in coordinating perceptual categorizations, both in multimodal relations and sequentially, over time. That is, the compositional process has two dimensions, first *laterally* in relating diverse sensorimotor categorizations and second *temporally*, as a nesting of network activations (so a currently active network is included within or *subsumed by* the network which is activated next).

A fundamental theoretical claim is that perception, meaning (conceptualization), and motor actions arise together, neither serially (perception flowing into action), nor in parallel (assembling results from independent modules) (Dewey, 1896). Furthermore, we do not know how to replicate this aspect of brain functioning computationally (but see Freeman (1991) for a rudimentary effort using chaos modeling). Indeed, the very role of the intermediate, ICS level of modeling is to provide a specification that moves us from our current inadequate understanding of brain function (where a symbolic cognitive model was assumed to be isomorphic to the brain mechanism) to an understanding of a new kind of physical process, that so far as we know only occurs today in biological processes (Edelman, 1992).

To achieve a parsimonious explanation of cognition, we will assume that there is fairly direct mapping between hierarchicalization of subgoals, impasses, operators, and chunking and the activation and relation of neural structures. Crucially, the architectural description we seek will not be constructed out of structures that cleanly map onto domain terms and relations. That is, there is no simple mapping between words and concepts (as characterized by Edelman). Rather, this non-linguistic processing involves recurrent relations between physical (neural) structures which constitute the perceptual and sensorimotor correlations of past neural activations. We shift from a stored structure manipulation view to a process construction view.

Fundamentally, we need the ICS account to explain not only "where the knowledge in the lower problem spaces comes from," but how the operator hierarchy is constructed, and even why operators are situation-action associations. As I have said, the ICS model presented here is based on Bartlett's model of remembering, applied to diverse examples, and Edelman's model of learning. My present purpose is to not to explain in detail where the ICS model comes from, but to apply it to Neomycin and Soar, in effect explaining why these cognitive models work.

Neomycin-Prime: Bridging The Gap Between Grammatical Models And Perceptual Recategorization

In this section, I will give a quick tour of Edelman's model, and work bottom-up to make connection with the analysis of Neomycin and Soar. Continuing with Edelman's presentation, we now consider the neurological underpinnings of conceptualization:

> What brain operations give rise to these properties? The TNGS [Theory of Neuronal Group Selection] suggest that in forming concepts, the brain constructs maps of its own activities, not just of external stimuli, as in perception. According to the theory, the brain areas responsible for concept formation contain structures that categorize, discriminate, and recombine the various brain activities *occurring in different kinds of global mappings*. Such structures in the brain, instead of categorizing outside inputs from sensory modalities, *categorize parts of past global mappings* according to modality, the presence or absence of movement, and the presence or absence of relationships between perceptual categorizations....
>
> Structures able to perform these activities are likely to be found in the frontal, temporal, and parietal cortices of the brain. They must represent a mapping of types of maps. Indeed, they must be able to activate or reconstruct portions of past activities of global mappings of different types—for example, those involving different sensory modalities. They must also be able to recombine or compare them. (Edelman, 1992; p. 109, emphasis added)

To understand this passage, we must understand Edelman and Reeke's (1988) model of categorization and the idea of a global mapping. The fundamental hypothesis in TNGS is that categorization is a process of establishing a relation between different neural structures, called *maps*. That is, categorization is a process of physically relating two maps, called a *classification couple*. Note that the emphasis is on physical activation of structures, not storage and retrieval. Hence, we talk of *categorization*, not categories per se. Furthermore, maps aren't merely isolated classifications of features, but always *relations* of classifications. Figure 11 (with Edelman's caption) introduces these ideas.

Each circle and square in Figure 11 represents a *neuronal group*, an intrinsically connected array of 50 to 10,000 neurons. Individual groups develop as the brain develops, but membership of neurons in a group develops through early sensorimotor coordination learning (called *developmental selection*), rather than being genetically pre-ordained or merely circumstantial correlations (like a palm print). Neural maps (referring again to Figure 11) in contrast, are entirely constructed and constitute the fundamental units of memory (called *experiential selection*). That is, a map is a set of neuronal groups locally related by sensory or motor signals (e.g., a visual system may have several dozen different maps, functionally specialized by color, movement, etc. (Edelman, 1992, p. 85)).

The formation of a map depends on the physical connections that grow when the brain develops, including of course physical links to sensory organs and muscles, and the relation to other mapping processes. In particular, functional segregation arises through local connections, as well as categorization learning (a process of differentiation between maps). Indeed, the point of Figure 11 is to show that activation of neuronal groups within a map is dependent on signals from correlated maps, not just sensory or motor signals. In a simple way, this diagram illustrates Dewey's (1896) criticism of serial, stimulus response theories: Each activation is part of a circuit, a coordination. We do not have temporally first the stimulus and then the motor response, but a single circuit of activations arising temporally together. As Edelman puts it, "memory is a property of the entire system" (p. 103), not a place where stuff is stored.

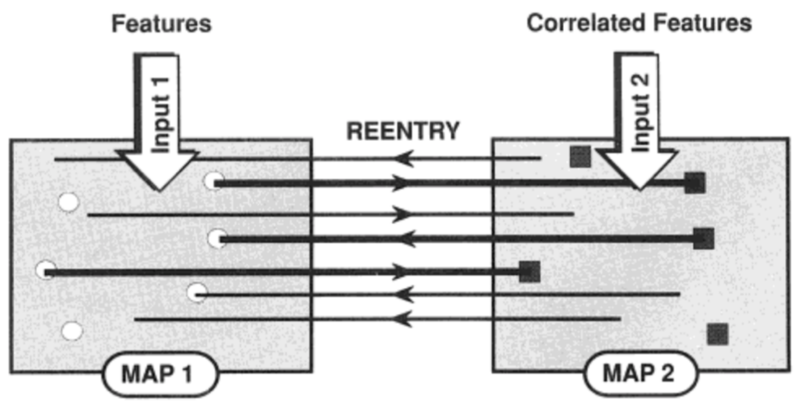

Figure 11. A classification couple, from (Edelman, 1992, Figure 9-4, p. 90).

Reentry. Two maps of neuronal groups receive independent inputs (1 and 2). Each map is functionally segregated; that is, map 1 responds to local features (for example, visually directed angles) that are different from those to which map 2 responds (for example, an object's overall movement). The two maps are connected by nerve fibers that carry reentrant signals between them. These fibers are numerous and dense and serve to "map the maps" to each other. If within some time period the groups indicated by the circles in map 1 are reentrantly connected to the groups indicated by the squares in map 2, these connections may be strengthened. As a result of reentrant signaling, and by means of synaptic change, patterns in responses in map 1 are associated with patterns of responses in map 2 in a "classification couple." Because of synaptic change, responses to present inputs are also linked to previous patterns of responses.

A *global mapping* involves categorization of maps; that is, a global map is a map of types of maps. This idea is illustrated in a highly abstract way by Figure 12.

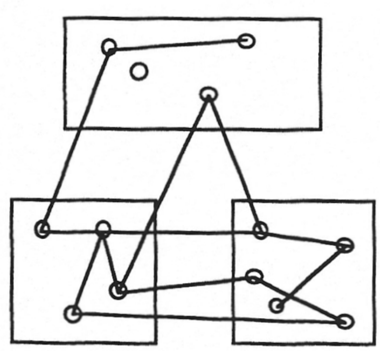

Figure 12. A map of maps (Adapted from Edelman, 1992; p. 103)

As mentioned in the opening quotes of this section, the brain categorizes its own activity, so signals in a map of maps come from other maps, including ultimately perceptual and motor classifications. A *global mapping* is a dynamic complex of such maps of maps, involving both sensory and motor maps, plus parts of the brain that are not mapped, such as the hippocampus, cerebellum, and basal ganglia, which are responsible for timing and sequencing (p. 105). In other words, a global mapping is Edelman's term for what Dewey called "a circuit." To be clear, a global mapping consists of many maps of maps like that shown in Figure 12.

Finally, as cited above, conceptualization is the process of *categorizing global maps*. As for all neural activation, conceptualization occurs—to put it quite simply—while everything else is happening. That is conceptualization occurs within an ongoing sequence of sensorimotor coordination (again, a fundamental claim made by Dewey in 1896). Here we find, in neuropsychological terms, the root idea of situated cognition: what we perceive, mean, and do arises together. In particular, what we *say*—our articulation of beliefs, models, and plans about the world and our activity—arises within our ongoing movement. Perception is not an isolated peripheral device that feeds input to a pattern matcher that selects descriptions and procedures from memory, to be subconsciously assembled into motor sequences. In terms of computational jargon, our memory is highly proceduralized, but *remembering is acting*, not an internal module or independent subprocess. Functional modularization occurs, but only within the context of other functional categorization, and this only within ongoing sequences of activity.

Two other points need to be emphasized before we leave this presentation of TNGS. First, all categorization occurs with respect to (in correlation with) internal criteria of value (p. 90), involving innate regulatory and survival functions. What we perceive is constructively biased by our prior conceptualizations and ongoing activity. This is another way of questioning the symbolic theory of cognition, which assumes that symbols are independent inputs, selected or filtered from the environment (Reeke and Edelman, 1988).

Second, the TNGS is being experimentally tested and refined in a series of programs and robots called Darwin. Edelman and Reeke claim that the robot NOMAD, which incorporates the Darwin III global mapping processes, is the "first non-living thing capable of 'learning' in the biological sense of the word" (p. 193). Illustrating the notion of a complete circuit of activation, "perceptual categorization occurs only when, after disjunctive sampling of signals in several modalities (vision, touch, joint sense), Darwin III activates an output through its reentrant maps." (p. 93). Edelman says that Darwin III "categorizes only on the basis of experience, not on the basis of prior programming." This means that the association between built-in sensory detectors (e.g., bumpy, striped) and movement is constructed in the global mapping, not stored as schemas or rules. At the lowest level, values relating to light and visual centers lead the robot to sample its environment and to target objects for contact.

From The Theory Of Neuronal Group Selection To Soar And Neomycin

The TNGS is a complex theory in its early stages. The Darwin-based robots accomplish only the most rudimentary forms of coordination. An evidently new kind of learning mechanism, based on selection and composition of sensorimotor maps, has been demonstrated. But Edelman himself is pessimistic that we will soon replicate the capabilities of the brain. For example, the Darwin model does not show how temporal sequencing and "dialectic" assembly is possible. Research on self-organizing systems is in its infancy. My specific goal here is to establish a link between the idea that conceptualization is a form of coordination and our existing cognitive models of knowledge and learning. This bridge can help neuropsychologists know what to look for, as well as provide new ideas for psychological experimentation and modeling.

Neural map activation over time

The first step is understanding how map activation occurs over time. Figure 13 illustrates how a given set of maps are reactivated at a later time. With similar inputs, there is similar output. The groups and the reentrant links that are activated are always prone to change, because of changes in the external environment and changes in the correlations activated throughout the neural system. Broadly speaking, TNGS claims that every activation is a *generalization* of past activation relations. Put another way, every activation is a *recategorization*, as opposed to a literal match or retrieval operation.

Now, this recategorization is always occurring simultaneously within a larger sensorimotor coordination, involving conceptualization (at the level of maps of types of maps) and, often in people, verbalization. A difficult concept for most AI researchers to grasp is that *activity* is defined broadly to include all ongoing ways of behaving, including sitting alone in a chair, silently talking to myself, visualizing, etc. Whether I am awake or dreaming, some activity is occurring all the time.

One of the most important aspects of behavior, emphasized by the neurologist Head, an English neurologist working in the early part of this century and a teacher of Bartlett, is the inherent sequential nature of learning. Head introduced the term "schema" in this context. In 1920, he wrote:

> Every recognizable change enters into consciousness already charged with its relation to something that has gone before.… For this combined standard, ...we propose the word "schema" …. Every new posture of movement is recorded on this plastic schema, and the activity of the cortex brings every fresh group of sensations evoked by altered posture into relation with it." (pp. 48-49, quoted by Rosenfield)

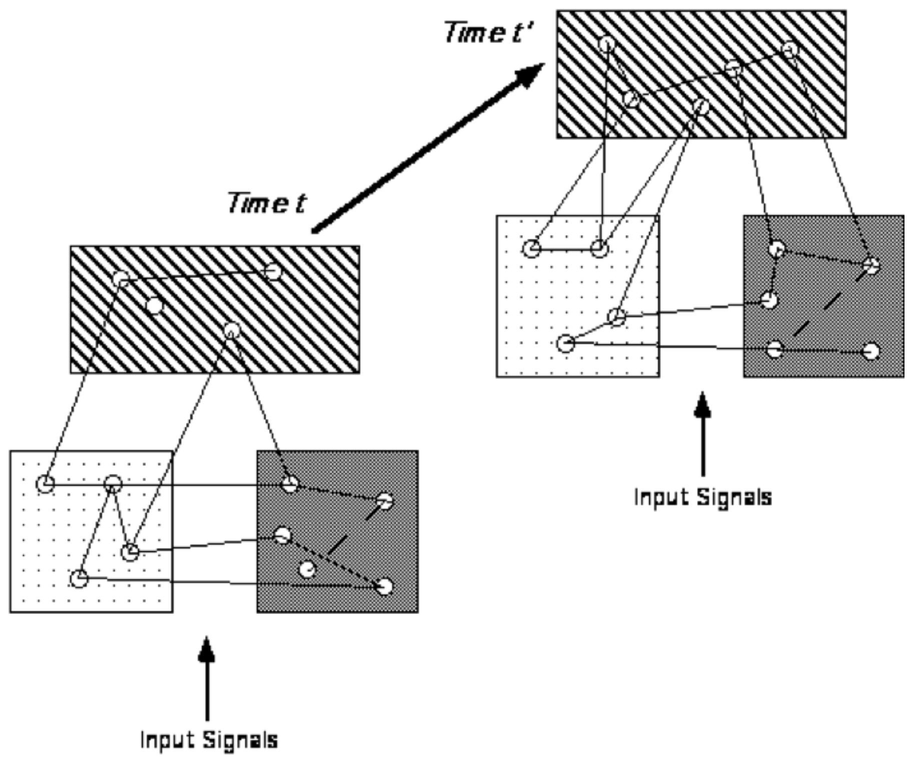

Figure 13. Memory as recategorization: Inputs categorized similarly produce similar output relations. (Adapted from Edelman, 1992; Figure 10-1, p. 103)

That is, besides memory for similarity, we have a *memory for sequence of activation*. By conjecture, this memory involves one map of maps activating another over time (Figure 14). Bartlett's notion of a schema is a *coordinating process*, rather than a stored description of how behavior appears to an observer, a characterization so common in the cognitive science literature:

> 'Schema' refers to an active organization of past reactions, or of past experiences, which must always be supposed to be operating in any well-adapted organic response. That is, whenever there is any order or regularity of behavior, a particular response is possible only because it is related to other similar responses which have been serially organised, yet which operate, not simply as individual members coming one after another, but as a unitary mass... All incoming impulses of a certain kind, or mode, go together to build up an active, organised setting: visual auditory... at a relatively low level; all the experiences connected by a common interest: in sport, in literature, history,... at a higher level... (Bartlett, 1932, p. 201)

Bartlett describes a schema as a chronological ordering of experience:

> If Head is right, ‘schemata’ are built up chronologically. Every incoming change contributes its part to the total ‘schema’ of the moment in the order in which it occurs. That is to say, when we have movements a, b, c, d, in this order, our ‘plastic postural model’ [in Head’s terminology] of ourselves at the moment d is made depends, not merely upon the direction, extent, and intensity of a, b, c, d, but also upon the chronological order in which they have occurred...

Schemas are *activities*, not stored and reinstantiated descriptions like semantic networks or production rules:

> Since its nature is not of that of a passive framework, or patchwork, but of an activity, it can be maintained only if something is being done all the time. So in order to maintain the ‘schema’ as it is—though this is rather inaccurate language—*a, b, c, d* must continue to be done, and must continue to be done in the same order. (p. 208)

This model strongly parallels Dewey’s (1938b) principles of *continuity* (developing sense of “what I am doing”) and *interaction* (coordinated perceptions and actions in the world):

> The two principles of continuity and interaction are not separate from one another. They intercept and unite. They are, so to speak, the longitudinal and lateral aspects of experience. Different situations succeed one another. But because of the principle of continuity, something is carried over from the earlier to the later ones. As an individual passes from one situation to another, his world, his environment, expands or contracts. (p. 519)

Dewey emphasizes that a new coordination necessarily requires a new way of categorizing the world. These two dimensions are readily apparent in Figure 14 (which I have developed based on the work of Dewey, Bartlett, and Edelman): Continuity is captured by the bold diagonal, signifying activation over time; Interaction is captured by the classification couples that correlate perception and action within this ongoing coordination.

In this figure, we see *different* maps of maps linked to *different* high-order maps. A perceptual detail may be a classification of a sound, a color, an object, a word, etc. The higher-order maps are physically linked by reentrant connections. However, the second higher-order map (labeled “time t’”) only becomes activated after the first map has become active (labeled “time t”). In this manner, it is conjectured that *maps of maps are sequenced over time via physical subsumption.* By conjecture, the number of possible connections in the brain is so large that there are sufficient links to establish a link between any sets of maps, as required by activity. Furthermore, all learning is categorization, in the form of the classification couple. This means that all learning is activation “in place” of pre-existing links, which are composed over time into, and always within, sequences of activation. As Edelman states, there must be some way of holding such activations so they can be compared and further coordinated, allowing for continuity in our experience.

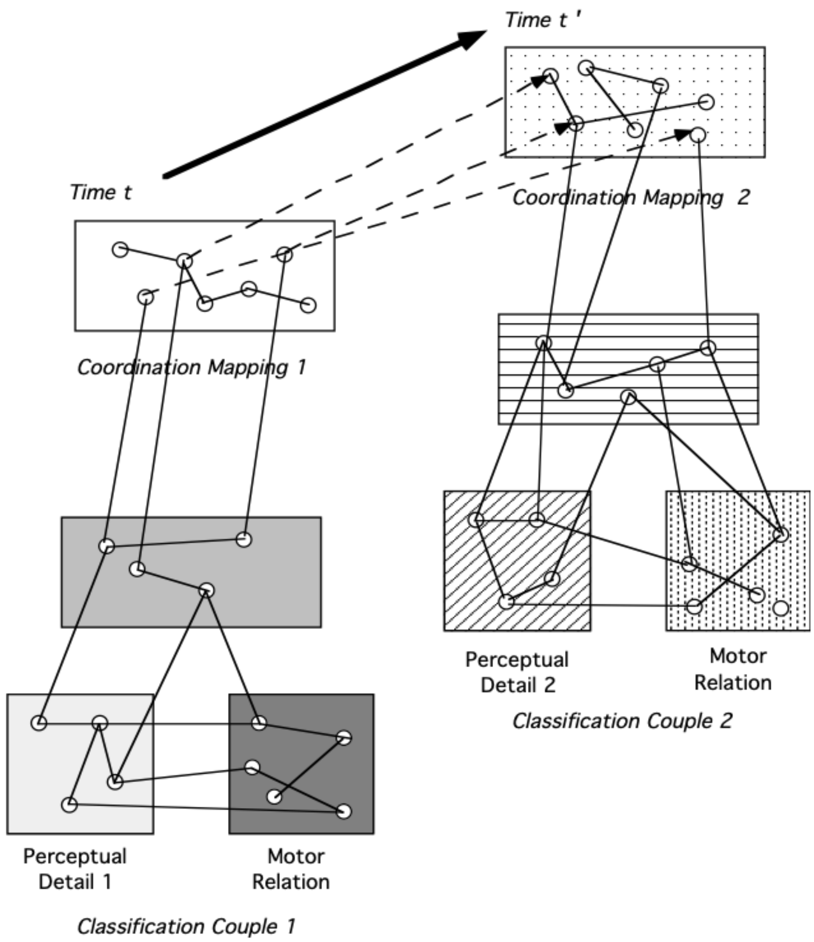

Figure 14. Higher-order maps, including categorizations of global maps, physically activate in sequence over time, serving as the memory of how perceptual details are coordinated within an activity (hypothesized extension of TNGS).

A thorough presentation of evidence for this model is beyond the scope of this paper. For further details, see my discussion of Dewey and Bartlett (Clancey, 1992c). See my review of *Bright Air, Brilliant Fire* (Clancey, 1993a) for a discussion of the nature of consciousness.

The ICS Notation Related To Neomycin

Diagrams of classification couples are a bit unwieldy. An alternative notation combines the maps of maps (rectangles) and secondary classification (e.g., motor relations) into a single node (Figure 15). Emphasis is thereby given to the sequence of perceptual details in the person's experience over time.

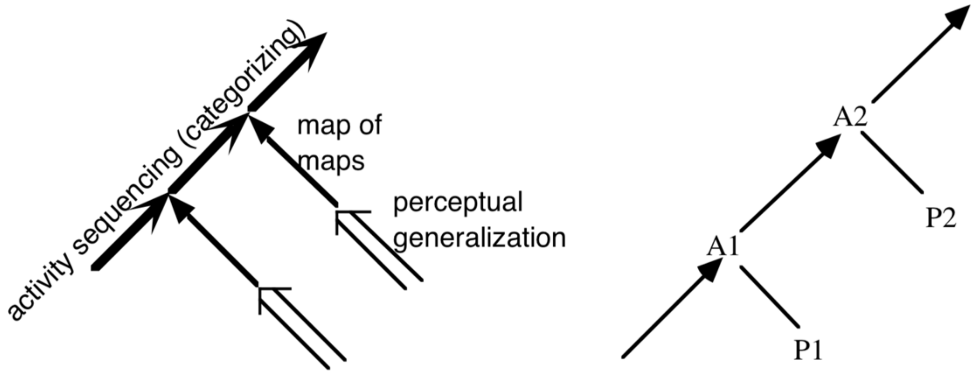

Figure 15. Basic "Interactive Control Structures" (ICS) diagram: Ai represent categorizations of global maps; Pi represent perceptual details.

Each sequence node (e.g., A1) of the ICS represents a conceptualization of global mapping. Below is shown the perceptual detail included in the associated global-mapping activation. This detail is the person's focus of attention, something consciously entertained. The next node (e.g., A2) is activated in sequence, over time. This conceptualization and associated global mapping is partially activated by the new environment, here represented exclusively by P2, another perceptual detail. Crucially, A2 is also activated by physical, reentrant links from the preceding global map, A1 (compare to Figure 14). In people, conceptualization is associated with linguistic processes, so the person could say "what I'm doing now." In using the ICS notation to represent problem-solving protocols, we think of each transition (e.g., from A1 to A2) as being a shift in "what I'm doing now," accompanied by a shift in perceptual detail. More generally, the ICS notation shows simply *a sequence of global mapping activations*, which in practiced form would not be accompanied by verbalizations (e.g., as in playing a musical instrument). How perceptual details in a sequence are related is a key part of the ICS theory.

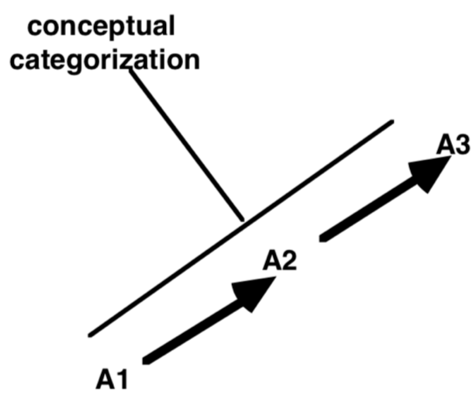

Figure 16. Conceptualization as categorization of sequences of activation.

Classification of such sequences of activation (Figure 14) is itself a form of conceptualization, corresponding to creation of a subgoal hierarchy (Figure 16; perceptual details are omitted here). The categorization itself enables the capability to "activate or reconstruct portions of past activities of global mappings of different types—for example, those involving different sensory modalities. They must also be able to recombine or compare them" (Edelman, p. 109). For example, my conceptualization of "Test-Hypothesis" includes speaking a name for the idea (itself a complex perceptual-motor coordination), a visualization of the name in a tree diagram (Figure 1), and the first step of checking for exposure to a disease.

Figure 17 shows Neomycin's subgoal hierarchy as a composition of conceptual categorizations. Primary transitions between global mappings are shown as bold arrows; categorization into higher-order maps is shown as left-slanting T bars. For example, the task Consult is a conceptualization of the sequence Make-Diagnosis and Prescribe-Therapy. The focus of subtasks are perceptual details, not shown here.

Moving in the other direction, the effect of practice is to establish reentrant links between perceptual details themselves, so attention to conceptualization drops out. In other words, a sequence of bindings is established from higher-order maps down to perceptual details and motor actions. For example, in Figure 16 this corresponds to a global mapping that connects Consult to Make-Diagnosis to Collect-Information to Establish-Hypothesis-Space, on down to finding out (FO) a particular finding for testing a particular hypothesis. That is, *the subgoal hierarchy is built up as categorizations of conceptualization sequences; chunks are established as categorizations of sequences of sensorimotor details*. I emphasize that the ordering dependency represented by the ICS is *temporal*. The hierarchical order of Figure 17 represents temporal dependency of activation, not a simple spatial property of physical inclusion of maps.

Figure 18 shows hypothetically how a sequence of operator activations becomes categorized as a chunk. A chunk is a sequence of activations of the details themselves, previously "bound" to a sequence of operators, proceeding from more general operators to more specific. Operands and "given" at a larger grainsize proceed to actions for transforming or relating them. The sequence of "bindings" is *one chunk*, which is modeled in production rule models by a new single production rule that combines the previous sequence of operators into one step. Although we describe a chunk as being a sequence of details, keep in mind that the chunk is created by a *categorization* of recurrent sequences of details.

"Givens" include what is immediately perceivable (but of course all details are categorizations, not objective stuff in the world); actions are themselves learned procedures for manipulating the environment (what to say, write, move, etc.). Operands get more specific as we go up the sequence of activation. This sequence, from general operator to more specific, "going forward" in the problem solving process, involves elaborating and relating details as we move into lower problem spaces. Finally, an *impasse* is the inability to continue an activation (ICS) sequence in an automatic way, requiring a physical recoordination coupled to perceptual refocusing.

More Details About Operator Conceptualization

To summarize, the ICS notation shows how operators are sequenced within a problem space (Figure 17). The process of categorizing these sequences produces a hierarchy of subgoals and their attendant problem spaces. These two dimensions of classification are shown by Figure 17: The problem spaces are the bold arrow sequences slanting to the right; the subgoal hierarchy is the composition of these sequences, shown as T-bars slanting to the left. The illustration of NL-Soar (Figure 9) shows the same hierarchical organization of operators, their sequence within a problem space, and how a given operator as a map of maps coordinates moving to another sequence of operators when an impasse occurs (shifting attention to the front of the sequence).

It is perhaps easy to see how categorization of perceptual activation sequences, corresponding to chunking, would produce a situation-specific production, essentially traversing an *existing* composition of operators and problem spaces. But how does categorization produce a new operator and the resulting operator hierarchy?

A clue was presented earlier when we observed that each operator in a Neomycin problem space (Figure 10) is at the same grainsize. That is, conceptualization of a sequence of *operators* occurs when the operators all *focus on the same kind of operand*. Referring to Figure 14, if classification couple #1 is reactivated or held active so it can be compared to classification couple #2, then a categorization will occur like that shown in Figure 13. That is, the perceptual details (foci) of the two operators will be seen as similar. The sequencing would then be maintained by different higher-order maps (as shown in Figure 14). The result is a sequence of activations with similar foci, corresponding to a problem space. By continued composition, a subgoal hierarchy of *abstract operators* will be constructed (Figure 17). Simultaneously, situation-specific sequences (chunks) will be constructed, cutting across levels of this hierarchy.

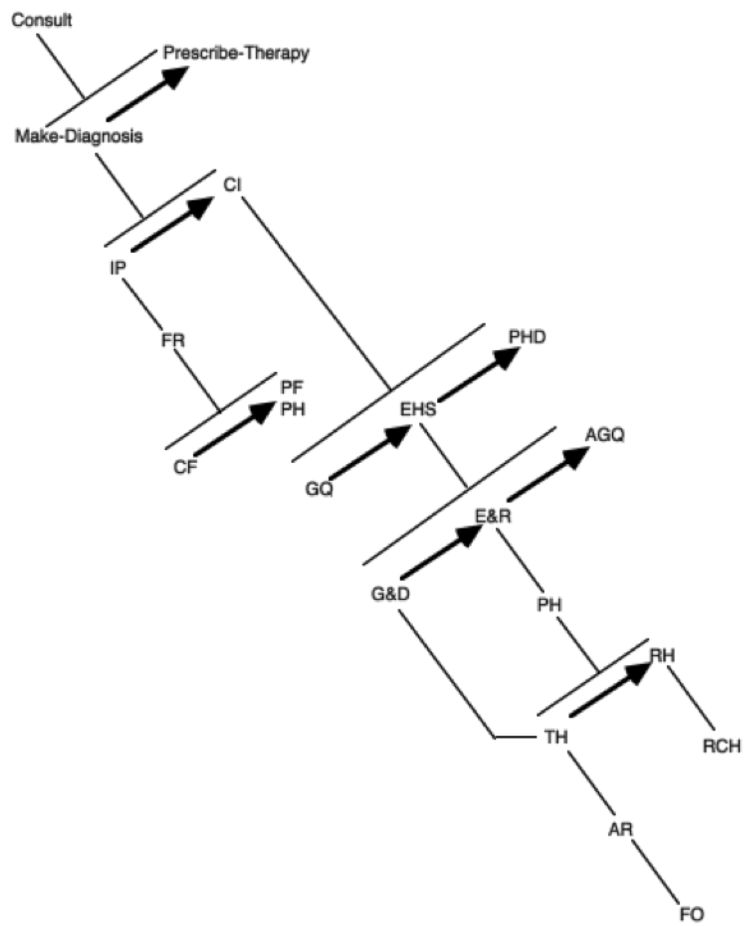

Figure 17. Neomycin subtask hierarchy shown as composition of sequences.

The question remains how and when operands are classified as similar. Certainly, return to a higher problem space after an impasse is an occasion to consciously notice the focus grainsize. For example, in diagnosis we might have been focusing on a particular hypothesis, and having completed that, recognized that we are returning to the level of selecting a hypothesis to focus upon. That is, we have returned to the differential, and this reactivation is categorized (or recognized) as the operator Explore-and-Refine. Methods (operators, aka metarules) are selected on the basis of this grainsize of interest. Put another way, perceptual segmentation of working memory or the environment activates alternative actions. Hence, it's not just what's in "working memory" that drives our reasoning, but "how we are seeing," our *manner of looking* (a point elaborated by Dewey (1896) and illustrated by Schön (1987)). Recall again the metaphor of looking up and looking down in searching a disease hierarchy (Figure 4).

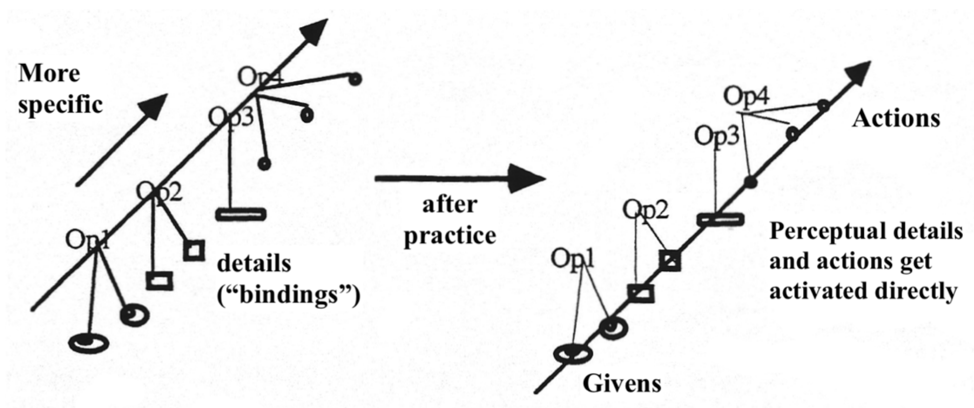

Figure 18. A chunk is a learned sequence of perceptual detail and motor activations.

Following the model of memory in Soar, no subgoaling occurs when direct recognition is possible. This suggests that subgoal composition only occurs when a breakdown forces a new problem space to be constructed (or reactivated). Such a top-down model makes sense if we consider how a student moves to more-specific focus of attention in the "what I'm doing now" hierarchy from "diagnosing a patient" to "focusing on a hypothesis, etc. Certainly much of our learning about complex systems is based on learning about an object and then learning about its properties and subcomponents (paralleling the idea of finer grainsize in a problem space hierarchy). On the other hand, movement from details to more complex constructs is also possible, suggesting that we also develop an operand hierarchy from an ontology.

The development of Neomycin did follow these two paths: Subtasks were defined on the basis of transformations required in the SSM, but new metarules tended to require a new domain relation (see "Learning new relations and subtasks"). I illustrate this development process as follows:

new problem space <— new focus grainsize
(subtask)
new domain relation <— **new operator** (metarule)

I had long ago detected this second (relation/operator) correlation, but only now understand, through the relation to NL-Soar and the ICS notation, how problem spaces (the subtask hierarchy) might be constructed. Of course, this example of development is our conscious process of designing Neomycin's procedures; it remains to gather evidence that such constructions parallel learning to speak. The ordering does fit the general notion that interaction in the world (attention to the SSM) motivates learning a new vocabulary term or relation, oriented around transformational activity.

To summarize a few claims of this model of learning:

- unlike Soar, perceptual recategorization occurs dynamically within every coordination; symbols are not given, but recategorized (addressing the "new term problem").
- as in Soar, comprehension, conceptualization, and word recognition would occur in parallel, in a single step, through practice (composing the traversal of different problem spaces).
- the model of learning is strongly bottom up (creating new operators from sequences and chunking), driven by and operating on perceptual details (i.e., ways of seeing are coupled to ways of doing).
- grammars and subgoal hierarchies are constructed and dynamically recategorized through an automatic process of sequencing and conceptualization, based on the ideas of classification couples, global maps, and pre-linguistic conceptualization.
- a problem space can be conceived as a sequence of perception-action activations, with similar perceptual details (i.e., operationalization is strictly tied to perceptual categorization of the SSM).

Conclusions

Obviously, a great deal more needs to be said about the Soar model and neural-level models. For example, NL-Soar uses a "snip" operator to undo a link in the SSM, as a simple form of backtracking. We would need to carefully work through the Darwin model of TNGS to see whether it enables this kind of working memory. In general, psychological evidence suggests that sequences are one-way (e.g., see Bamberger's (1992) study of music), so we cannot arbitrarily move around within a problem space. But what aspects of conceptual recoordination are possible? How do these differ between humans and primates and other animals?

The Neomycin-Soar effort uncovered many of the connections between the two programs I have summarized here (Table 1). However, the relation between the SSM and the levels of the problem space hierarchy has not been noted elsewhere, to the best of my knowledge[7]. It appears that NL-Soar was the first program in the Soar series to describe problem solving in terms of operators for constructing a model.

Neuro-Soar (Cho, et al., 1991) is an initial "connectionist" implementation of Soar's production memory and binding process. But impasse detection, subgoaling, and learning were omitted. Models like TNGS suggest that efforts to develop new functional architectures should instead begin with categorization and attempt to show how subgoals are conceptualized.

The analysis given here has direct implications for instructional design. For example, in developing Guidon-Manage (Rodolitz and Clancey, 1989) we were concerned with teaching the diagnostic procedure of Neomycin. It is now evident that, heuristically, learning new operators should be grounded in an understanding of the conceptual levels of the SSM (Figure 10). In related work with students graphing lines on a computer screen (Clancey, 1994), we found that not knowing where to look or what grainsize is significant (e.g., pixels, bundles of lines) produced great ambiguity in interpreting the lesson materials.

Most of all, this analysis confirms the basic organization of today's cognitive models. Indeed, in showing how Neomycin and Soar have basically the same architecture (excluding chunking), and then in relating the problem-space hierarchy to Edelman's TNGS idea of conceptualization and classification couples, we have considerably tightened up the case for using such levels of abstraction in student modeling. On the other hand, we have begun to discern points of inflexibility, ways in which the exclusively language-based model will break down. The analysis given here (particularly Figures 14 and 17) suggests that problem spaces are not merely applied when an impasse occurs, but dynamically constructed by activating previous coordinations. This suggests that in so far as the perceptual categorizations are stable, the operators and hence subgoal conceptualization will be stable. On the other hand, in so far as the problem solver is seeing a problem in a new way (e.g., reconsidering how causal processes might in general interact)—effectively reconceptualizing the grainsize and nature of focus relations—the nature of a problem space will change. It is this ability to perceptually recategorize the world, detect new kinds of similarities, and compose new hierarchical sequences that Soar lacks. TNGS, as extended by the ICS notation here, begins to explain how linguistic terms are grounded, new meanings are constructed, and subgoals organized into hierarchies coupled to perceptual categorizations.

Proceeding from here, we might look at developmental data in instructional settings in a new way. First, protocols that don't fit the purely symbolic-linguistic approach might be explained in terms of the reconceptualization occurring as the problem is being solved, which I have described. We can model this reconceptualization in terms of words, definitions, and schemas, but according to situated cognition and the TNGS more specifically, there will always be some residual, non-patterned aspects to behavior. Second, we might pursue new forms of experimentation that are directed at manifesting perceptual recategorization and its relation to operator construction. By the idea that perception and action are coupled, arising together, we might find ways in which new possibilities for transformation, in the form of artifacts and tools in general, might stimulate or enable perceptual, and hence linguistic, change.

---

[7]Hobbs (1985) does emphasize the nature of granularity in relation to theories: "When we move from one level of a hierarchy to the level below, we are moving from a coarse-grained local theory to a more fine-grained local theory, and the axioms that specify the decomposition of coarse-grained predicates into fine-grained ones constitute the articulation between the two theories." (p. 434)

References

Bamberger, J. (1991). *The mind behind the musical ear*. Cambridge, MA: Harvard University Press.

Bartlett, F. C. [1932] (1977). *Remembering--A Study in Experimental and Social Psychology*. Cambridge: Cambridge University Press. Reprint.

Cho, B., Rosenbloom, P.S., and Dolan, C.P. (1991). Neuro-Soar: A neural network architecture for goal-oriented behavior. *Proceedings of the Cognitive Science Society,* pp. 673-677.

Clancey, W.J. (1988). Acquiring, representing, and evaluating a competence model of diagnosis. In M.T.H. Chi, R. Glaser, and M.J. Farr (editors), *The Nature of Expertise*. Hillsdale: Lawrence Erlbaum Associates, pp. 343-418.

Clancey, W.J. (1991a). Interactive Control Structures. Unpublished technical report. To appear in *Situated Cognition: On human knowledge and computer representations*.

Clancey, W.J. (1991b). Review of Rosenfield's "The Invention of Memory." *Artificial Intelligence,* **50**(2):241-284.

Clancey, W.J. (1991c). Situated Cognition: Stepping out of Representational Flatland. *AI Communications—The European Journal on Artificial Intelligence* **4**(2/3):109-112.

Clancey, W.J. (1992a). Model construction operators. *Artificial Intelligence,* **53**(1) 1-124.

Clancey, W.J. (1992b). New perspectives on cognition and instructional technology. In E.Costa (editor), *New Directions for Intelligent Tutoring Systems*, Berlin: Springer-Verlag, pp. 3-14.

Clancey, W. J. (1992c). 'Situated' means coordinating without deliberation. Presented at the *McDonnell Foundation Conference on The Science of Cognition,* Sante Fe. Unpublished IRL Technical Report.

Clancey, W.J. (1993a). The biology of consciousness: Review of Rosenfield's "The Strange, Familiar, and Forgotten" and Edelman's "Bright Air, Brilliant Fire." Artificial Intelligence, 60(2)313-356.

Clancey, W.J. (1993b). Guidon-Manage revisited: A socio-technical systems approach. *Journal of AI and Education* , **4**(1)5-34.

Clancey, W.J. (1993c). Notes on Heuristic Classification. *Artificial Intelligence*, **59**(1-2)191-196, February.

Clancey, W.J. (1993d). Situated action: A neuropsychological interpretation (Response to Vera and Simon). *Cognitive Science*. **17**(1) 87-116.

Clancey, W.J. (1994). Situated cognition: How representations are created and given meaning. In R.Lewis and P.Mendelsohn (editors), *Lessons from Learning, IFIP Transactions A-46*. Amsterdam: North-Holland, pp. 231-242.

Dewey, J. [1896] (1981). The reflex arc concept in psychology. *Psychological Review*, III:357-70, July. Reprinted in J.J. McDermott (editor), *The Philosophy of John Dewey,* Chicago: University of Chicago Press, pp. 136-148.

Dewey, J. (1938a). *Logic: The Theory of Inquiry*. New York: Henry Holt & Company.

Dewey, J. [1938b] (1981). The criteria of experience. In *Experience and Education*, New York: Macmillan Company, pp. 23-52. Reprinted in J.J. McDermott (editor), *The Philosophy of John Dewey,* Chicago: University of Chicago Press, pp. 511-523.

Edelman, G.M. (1987). *Neural Darwinism: The Theory of Neuronal Group Selection*. New York: Basic Books.

Edelman, G.M. (1992). *Bright Air, Brilliant Fire: On the Matter of the Mind*. New York: Basic Books.

Freeman, W. J. (1991). The Physiology of Perception. *Scientific American*, (February), 78-85.

Hobbs, J. R. (1985). Granularity. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*. Los Altos: Morgan Kaufmann Publishers. pp. 432-435.

Iran-Nejad, A. (1987). The schema: A long-term memory structure or a transient functional pattern. In R. J. Tierney, Anders, P.L., and J.N. Mitchell (editors), *Understanding Readers' Understanding: Theory and Practice,* (Hillsdale, Lawrence Erlbaum Associates)

Laird, J.E. and Rosenbloom, P.S. (1990). Integrating Execution, Planning, and Learning in Soar for External Environments. *Proceedings of the Eighth National Conference on Artificial Intelligence. Menlo* Park, CA: The AAAI Press. pps. 1022-1029.

Lakoff, G. (1987). *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind.* Chicago: University of Chicago Press.

Lave, J. (1988). *Cognition in Practice*. Cambridge: Cambridge University Press.

Lehman, J.F., Lewis, R.L., Newell, A. (1991). Natural language comprehension in Soar: Spring 1991. Technical Report CMU-CS-91-117.

London, B., and Clancey, W. J. (1982). Plan recognition strategies in student modeling: Prediction and description. AAAI-82, pp. 335-338.

Newell, A. and Simon, H.A. (1972). *Human Problem Solving*. Engelwood Cliff, NJ: Prentice-Hall, Inc.

Pople, H. E. J. (1982). The development of clinical expertise in the computer. In P. Szolovits (editor), *Artificial Intelligence in Medicine* (pp. 79-117). Boulder, CO: Westview Press.

Reeke, G.N. and Edelman, G.M. (1988). Real brains and artificial intelligence. Daedalus, **117** (1) Winter, “Artificial Intelligence” issue

Rodolitz, N. S., and Clancey, W. J. (1989). GUIDON-MANAGE: teaching the process of medical Diagnosis. In D. Evans, & V. Patel (editors), *Medical Cognitive Science*. Cambridge: Bradford Books, pp. 313-348.

Rosenfield, I. (1988). The Invention of Memory: A New View of the Brain. New York: Basic Books.

Ryle, G. (1949). *The Concept of Mind*. New York: Barnes & Noble, Inc.

Sack, W., Soloway, E., and Weingrad, P. (1991). Re:Writing Cartesian student models. Technical report, University of Michigan.

Schön, D.A. (1979). Generative metaphor: A perspective on problem-setting in social policy. In A. Ortony (editor), *Metaphor and Thought*. Cambridge: Cambridge University Press. 254-283.

Schön, D.A. (1987). *Educating the Reflective Practitioner*. San Francisco: Jossey-Bass Publishers.

Schön, D.A. (1990). *The theory of inquiry: Dewey’s legacy to education*. Presented at the Annual Meeting of the American Educational Association, Boston.

Smoliar, S.W. (1989). Review of *Neural Darwinism: The Theory of Neuronal Group Selection*, by G.M Edelman. *Artificial Intelligence* **39**(1), 121-136.

Washington, R. and Rosenbloom, P.S. (1988). Applying problem solving and learning to diagnosis. Unpublished technical report, Knowledge Systems Laboratory, Stanford University.

Wenger, E. (1987). Artificial Intelligence and Tutoring Systems: Computational and cognitive approaches to the communication of knowledge. Los Altos: Morgan-Kaufmann.

Wilkins, D. C., Clancey, W. J., & Buchanan, B. G. (1988). On using and evaluating differential modeling in intelligent tutoring and apprentice learning systems. In J. Psotka, D. Massey, & S. Mutter (editors), *Intelligent Tutoring Systems: Lessons Learned,* Hillsdale, NJ: Lawrence Erlbaum Publishers, pp. 257-284.

Acknowledgments

Steve Smoliar and the members of the AI Lab of Zurich University provided helpful comments for revising this paper. Financial support is provided in part by Nynex Science and Technology, Inc.