

ARTINT 1017

Response

On *Unified Theories of Cognition*: a response to the reviews

Paul S. Rosenbloom

*Information Sciences Institute & Department of Computer Science, University of
Southern California, 4676 Admiralty Way, Marina del Rey, CA 90292, USA*

John E. Laird

*Artificial Intelligence Laboratory, 1101 Beal Street, The University of Michigan, Ann
Arbor, MI 48109, USA*

Received August 1991

Revised September 1992

1. Introduction

This is a tough one. Clearly, neither of us is Allen Newell, and there is no way that we can do the job he would have liked to have done in responding to these thought-provoking reviews of *Unified Theories of Cognition* (henceforth, *UTC*). However, it was very important to Allen—and is very important to us—that the work on Soar and unified theories of cognition carry on. So, here we will do what we can based on our combined understanding of Soar, cognitive science, Allen, and how he would have likely responded to the issues raised by these reviews. In this latter, we have been greatly aided by a prepublication print of [12] that was graciously provided to us by the editors of *Behavioral and Brain Sciences*, and by Newell's follow-on paper to *UTC* [13]. In general, we will try to be clear about when we are conveying how we think Allen would have responded, and when we are just giving our own views.

Correspondence to: P.S. Rosenbloom, Information Sciences Institute & Department of Computer Science, University of Southern California, 4676 Admiralty Way, Marina del Rey, CA 90292, USA. E-mail: rosenblo@isi.edu. Telephone: (310) 822-1511.

UTC has three principal themes:

- (1) "Psychology has arrived at the possibility of unified theories of cognition—theories that gain their power by positing a single system of mechanisms that operate together to produce the full range of human cognition. I do not say they are here. But they are within reach and we should strive to attain them" (p. 1).
- (2) There is a common foundation underlying cognitive science.
- (3) Soar is a candidate unified theory of cognition that is useful as an exemplar of the concepts introduced in the book.

The first theme is most contentious among traditional experimental psychologists. As none of the present reviewers fit this mold, it is not too surprising that this issue is not a major source of concern for them. Most of the reviewers range from neutral (i.e., they don't mention it) to strong support. The one exception is Minsky, who is "not sure that this is a good idea". However, our impression is that Minsky is really disagreeing with a stronger form of this theme than Newell actually intended. By "unified theory" Newell was referring to any theory that integrates together a set of mechanisms that are intended to model a broad swath of human cognition. Newell's (and our) approach to building such a theory is biased towards having a small set of mechanisms that produce most of the action through their interactions (an issue that we will get back to later). However, the call for the development of unified theories is not limited to just this form of unification. In particular, an integrated architecture built out of a, possibly large, set of specialized modules—such as the Society of Mind theory might ultimately lead to—seems well within the scope of the call.

Though the first theme was clearly the central one in *UTC*, Minsky's comments were the only ones about it, so we won't say anything more about it in the remainder of this response.

The second theme covers basic material on behaving systems, knowledge systems, representation, machines and computation, symbols, architectures, intelligence, search and problem spaces, preparation and deliberation, system levels, and the time scale of human action. Much of this material is common knowledge by now in cognitive science, though still by no means universally accepted (in particular by those more closely tied to the structure of the brain). The new material that might be expected to raise the most controversy comprises:

- (1) *The refinement of the notion of symbol and the resulting distinction between symbols and representation.* In Newell's usage, a symbol is a pattern that provides access to distal structures (so that local processing can make use of structures which are initially not localized).

Symbols are involved whenever such distal access occurs. A representation, on the other hand, is defined by encoding and decoding functions, such that if you decode the situation that results from applying an encoded transformation to an encoded situation, then you have exactly the situation that would have been generated if the original transformation had been applied to the original situation; that is:

$$\begin{aligned} & \text{Decode}(\text{Encode}(\textit{Transformation})[\text{Encode}(\textit{Situation})]) \\ & = \textit{Transformation}[\textit{Situation}]. \end{aligned}$$

Using a representation need not involve distal access, nor must distal access involve representation. Thus our interpretation of this distinction is that it should be possible to have representation without symbols and symbols without representation.

- (2) *The definition of intelligence in terms of approximation to a knowledge-level system.* This definition distinguishes the concept of intelligence from the concepts of knowledge, level of performance, and generality by defining intelligence *with respect to a goal* to be how well the *extant knowledge* is used in selecting actions for the goal. So, lack of knowledge implies ignorance, but not lack of intelligence. Failure to perform at a high level could result from a lack of intelligence, but could just as well arise from a lack of either the appropriate knowledge or goals. Generality, rather than being an inherent attribute of intelligence, becomes the scope of goals and knowledge possessed; and general intelligence becomes the ability to exhibit intelligence across a wide range of goals and knowledge. Note that though intelligence is defined in a universal manner, the intelligence exhibited by an agent need not be unitary, and can in fact be quite domain-specific because the ability to use knowledge effectively may not be the same across different ranges of goals and knowledge.
- (3) *Describing human action in terms of a hierarchy of system levels which are characterized by time scales roughly an order of magnitude apart.* For example, at 1 ms is the neuron level, at 10 ms is the neural circuit level, and at 100 ms is the deliberate-act level. These levels then aggregate into bands that share a phenomenal world. The biological band consists of three levels concerned with the physical structure of the brain: the neuron and neural-circuit levels, plus one lower level. The cognitive band consists of three levels concerned with symbol processing: the deliberate-act level plus two higher ones. The rational band consists of three levels concerned with intendedly rational behavior; that is, humans at these levels approach knowledge-level systems.

Many of the reviewers comments are focused on foundational topics, though not limited to just these novel contributions.

The third theme, not surprisingly, engenders a large number of comments from the reviewers. Even those who share the same assumptions on unified theories and foundations generally have quite different notions about what the right architecture is for cognition, and about how one ought to go about developing and studying it (including, in particular, what data should drive its development). This is in fact one of the major messages of *UTC*—that Soar is not being proposed as *the* right answer, but as an exemplar (both Purves and Arbib appear to overinterpret the claim being made about Soar), and that others should take up the challenge from their own perspectives (and with their own driving data) and develop alternative unified theories of cognition. However, it is also clear that with all of Soar laid out in *UTC*, it becomes an irresistible target for comment (and that this isn't really inappropriate). So, given that the reviewers go beyond the book's use of Soar, to fundamental comments on its nature, we will feel free to also go beyond the book in responding; in particular, we will refer to results generated since the book's completion wherever they seem relevant to discussions of the architecture and of how it is being developed and studied. In general, we will only provide references for work not already cited in *UTC*.

In the body of this response we focus on the principal comments on both of these latter two themes—foundations and Soar. Related analyses of some of these same issues, along with analyses of some additional issues, can be found in [8,16].

2. Foundations

Two of the three foundational themes that were expected to be controversial—(2) defining intelligence and (3) the bands (and levels) of human cognition—did indeed provoke the lion's share of comments, along with one topic that surprised us, though perhaps it shouldn't have: theories. These three topics are discussed in this section, along with two other topics that had fewer, but still significant, comments: natural language and evolution.

2.1. Defining intelligence

Fehling and Hayes-Roth have several questions about Newell's proposed definition of intelligence. We've already commented on the question of whether intelligence is a unitary quantity or whether intelligence can vary by domain. A second question is the extent to which intelligence is really independent of knowledge. Newell clearly meant that adding a new piece of

knowledge relating goals to actions doesn't increase intelligence. However, we don't think he also meant that there is no way to increase intelligence by adding meta-knowledge. It seems quite consistent with the basic definition to say that adding a body of meta-knowledge, *M*, that makes the system better able to use another body of knowledge, *B*, can make the system more intelligent *with respect to the uses of B*.

A third question about the definition of intelligence is whether a system that performs perfectly within a very narrow region (such as a thermostat) is more intelligent than a system that performs less perfectly over a wide region (such as a person); or, similarly, whether a system that performs perfectly via a trivial method (such as table look-up) is more intelligent than one that performs less perfectly via a more sophisticated method (such as reasoning from first principles). This is an interesting question that seems to reduce to three fundamental components: ignorance, generality, and complexity. First, if an agent doesn't know the first principles, then it may certainly be ignorant, but not necessarily unintelligent. Likewise, if it knows both a perfect table and an imperfect set of first principles, it is more intelligent with respect to those goals to use the perfect table in answering rather than the first principles. Second, what should make the first principles advantageous is not their substandard use for goals where the table is applicable, but their use for other goals where the table is inapplicable. Thus this is an issue of generality rather than intelligence. Third, there appears to be an implicit appeal to the notion that something that is more complex (or sophisticated) is more intelligent. It is often assumed that complexity is a sign of intelligence, but what must really be meant is the ability to cope with complexity when needed (we've all seen too many complex descriptions that could really have been expressed quite simply). Using the first principles where the table is applicable would be inappropriate complexity. However, if the system knew the principles, but could not use them when they might help with other goals, that would indeed be a failure of intelligence.

A fourth question about the definition of intelligence is whether it ignores important phenomena such as: "a system's allocation of limited cognitive resources, its ability to focus attention, its efforts to meet real-time constraints, its balancing competing demands and opportunities for action, ..." (Hayes-Roth). The answer to this question is that intelligence is about how closely a physical system can approximate the knowledge level; that is, the issue is how the lower levels can provide a rational band. The phenomena Hayes-Roth mentions are precisely about how a physical system can best reflect its knowledge and abilities in resource-limited situations. Thus, systems exhibiting these phenomena are likely to be judged more intelligent by the knowledge-level definition than systems that do not.

Rather than being directly concerned about the definition of intelligence, Schank and Jona are concerned about the definition of AI. They take an

eclectic approach that has more in common with Newell's list of constraints that shape the mind (p. 19) than with his proposed definition of intelligence. In Schank and Jona's view AI consists of four topics: learning, memory organization and access, functional constraints + knowledge analysis, and scale-up. All of these topics, except for knowledge analysis (which is discussed in Section 2.2.2), do appear to be covered in Newell's list. Schank and Jona also appear to leave important topics out, such as performance, reasoning, problem solving, planning, and natural language, but without a more detailed description of what they had in mind, it is hard to say much concrete about this apparent lack.

2.2. Bands and levels

Arbib has a general concern that the bands and levels described by Newell are too influenced by their roots in serial 1971-style computers, and thus generally inappropriate for describing human cognition. However, we don't see how a careful reading of the relevant figures in *UTC* could be taken as supporting this concern. Instead, it appears that Arbib is confusing the hierarchy of computer systems (Fig. 2-3, p. 47) with the time scales of human action (Fig. 3-3, p. 122). For example, though a register-transfer level appears in the former figure, there is no sign of it in the latter figure. Likewise, the symbol level in the latter figure—which occurs at $\sim\sim 10$ msec and is identified with the biological level of neural circuits—is in no way inherently serial (see, for example, how this level is mapped onto Soar in Fig. 4-23 on p. 224). We thus don't see how Arbib's concerns are relevant to the levels Newell describes in Fig. 3-3.

2.2.1. The cognitive band

The remaining concerns tend to be about Newell's focus in *UTC* on the cognitive band, with only limited attention being paid to the rational and biological bands (and even less to the others). This focus comes in for some sharp criticism from the reviewers who think it is either too restricted or completely misplaced.

In terms of the focus being too restricted, Arbib is concerned that an attempt is being made to restrict the scope of cognitive science to rational behavior, to the exclusion of everything else; and, in particular, to the exclusion of subjects such as instinctive behavior. However, this seems to us to be a misreading of *UTC* in several ways. First, Newell clearly recognized that models of all of the bands were necessary for a complete model of human behavior. Second, the rational band is the home of rationality, not the cognitive band. In the cognitive band there can easily be procedures which can be executed, but not examined, thus providing "knowing how" without "knowing that". For example, Soar is "aware" of the contents of its working

memory, but it has no way to directly examine its productions—indeed, it can only execute them. Instinctive behavior could thus be represented at the cognitive band as unexaminable proceduralized content (such as productions) that is innate and fixed, or indeed simply as part of the architecture.

In terms of the focus being misplaced, Schank and Jona, and Arbib, are concerned that too much attention is being paid to the cognitive band and too little to the rational or biological bands (respectively). Fehling and Purves appear to go beyond this concern to a claim that only one band matters, though they are in complete disagreement as to whether it is the rational (Fehling) or biological (Purves) band. These concerns and claims about the misplacement of the focus are considered in more detail next.

2.2.2. The rational band

Fehling believes that the utility in models of cognition comes from competence theories, and that one key aspect of this is the utility of modeling knowledge rather than mechanism. This aspect is clearly a variation on the expert-system slogan that “Knowledge is Power”, as are Schank and Jona’s comments that “. . . the important thing to study, if one wants to understand the nature of intelligence and build intelligent systems, is content not architecture.” and “Knowledge is what determines the architecture.” We can’t disagree with the importance of explicitly studying knowledge, nor with the claim that this has not so far been a central scientific component of the Soar research enterprise. There has been some focus on knowledge, particularly in situations where the functionality being provided is both particularly important and provided in a non-obvious manner (such as when it provides a new form of learning behavior). There has also clearly been a significant amount of informal study of knowledge in the context of applying Soar in various domains. Nonetheless, these activities have not been a large part of the overall enterprise.

Though we do expect the explicit study of knowledge to be a continually growing component of the Soar research effort, this should not be thought of as taking away from the central importance of studying the architecture. The human cognitive architecture is the fixed structure that makes the human mind what it is. It distinguishes people from other animals, and enables people to acquire and effectively use the wide range of knowledge that then lets them perform effectively in such a diversity of domains. In natural systems, at least, the architecture (plus the world and the already extant knowledge) determines what knowledge the agent can and does have, not the other way around. In general, without an architecture, knowledge is nothing more than useless scribbles, and without an appropriate architecture, intelligence (and thus performance) can be severely degraded.

Fehling also believes that a clean separation should be maintained between mind and brain (rather than attempting to ground mental constructs in the physical structures of the brain). He is therefore concerned that Newell is requiring all cognitive theory to be grounded in the physical structure of the brain, and that this is unworkable, unnecessary, and inappropriate. However, here we can only assume that Fehling has radically misunderstood *UTC*. The notion of a system level implies that it is possible to build theories at various levels of the hierarchy without attending to the details of lower levels. Such theories can, in fact, be extremely useful in their own right; a fact that Newell was well aware of, and himself took advantage of throughout his career. Thus there is definitely no intent in *UTC* to claim that all cognitive theories must be so grounded. However, to the extent that the desire is to get an understanding of human cognition at all levels and/or the extent to which the level boundaries are not hard, a one-level (or one-band) model is insufficient. This is not particularly pernicious nor behaviorist—it freely allows the use of “mental” terms—it just hopes to eventually understand how they are grounded, no matter how indirectly or complexly, in the lower levels/bands. What Chapter 3 of *UTC* represents is an initial attempt by Newell to help lay some of the groundwork for bridging the gap between the cognitive and biological bands (see [1] for a small additional step in this direction). There is a very long way to go here still, but the problem has to rank as one of the most important and challenging problems in modern science.

2.2.3. *The biological band*

Purves' primary concern with *UTC* is “its consistent failure to deal with the essence of cognition, that is, the brain and what is presently known about it.” The strong version of this claim is that neurobiological evidence is primary, and that nothing of true importance can be done without it. We find such a claim hard to take seriously, for two reasons. The first reason is that equivalent claims can be made by all bands with respect to those above them; for example, physicists could just as easily claim that chemists should stop speculating about how the world works, and “put on their overalls, open the hood, and take a look”; and chemists could make the same claim about biologists. The fallacy, of course, in all of this is that each band talks about different phenomena and captures different regularities, and thus has its own appropriate technical languages and models. Understanding the relationships between the bands is ultimately of central importance to science; however, it can't replace studying each of the bands in its own right. The second reason is that ultimately the proof is not in such argumentation, but in what research paths lead to understanding cognition. To take one example from *UTC*, is Purves ready to claim that

neurobiology can, any time in the near (or even relatively distant) future, make zero-parameter predictions about the amount of time it will take a human to execute a computer command as a function of how the command is abbreviated? Or, to take an important example from elsewhere, can it tell us anything about how students learn, and fail to learn, subjects like mathematics? These two tasks stand in for a whole class of results from the cognitive (and, indeed, rational) band that have both increased our understanding of human cognition and have had significant practical impact. On practical impact, though the command-abbreviation result has not itself had significant economic impact, closely related results have—just one such result saved NYNEX \$2.4M per year [4]. More broadly, the whole area of expert systems is a spin-off from the study of the cognitive and rational bands.

The weak version of Purves' comment is that *UTC* is weaker than it would have been had it taken neurobiological data into account (a view which appears to match Arbib's). We are quite willing to believe that this might be true, as it is also likely true with respect to linguistic data and a variety of other bodies of data about human behavior. However, a significant breakthrough in bridging the gap between Soar and neurobiology will probably need to wait for the development of individual researchers who are sufficiently proficient in both of these topics.

Beyond Purves' general concern, he also has a more specific concern about the timings assigned to particular levels. He discusses how the brain can actually make discriminations based on time differences much smaller than the ~ 1 msec rate suggested for the neuron level; in fact, it seems to be able to discriminate intervals as short as ~ 10 μ sec. While we are impressed with the cleverness of how the brain appears to accomplish this, it is hard to see how it bears on the issue of the time scale(s) on which the brain functions. The leading model of auditory localization appears to be that the brain can respond differentially as a function of the correlation between the signals from the two ears [17]. The problem is that there need be no relationship between the time to compute this correlation and the size of the temporal interval that the correlation then implies. Thus, it is quite consistent for the brain to take milliseconds to compute these microsecond differences (though we have not actually seen any data on the exact time course).

2.3. Theories

Concerns about Newell's usage of theories includes the (lack of) methodology for studying and evaluating them, the question of their identifiability, the nature of theoretical form and content, and competence theories.

2.3.1. Methodology

Hayes-Roth is concerned with the development of a “rigorous methodology for describing, analyzing, and empirically evaluating complex cognitive theories”. We agree with her general call to arms, though realize that developing such a methodology will be a difficult and extended process. One subpoint worth additional comment is the relationship between a theory and its implementation. Based on the notion of theory development that Newell expressed, this issue, along with the related issue of when a theory should be adopted or rejected (Fehling), loses some of its force. To Newell, the implemented architecture was the theory, implementation hacks and all. The hacks are just part of the approximate nature of the theory. Since incorrectness of one part of a theory is not to be grounds for rejecting the whole theory, this amount of approximation is quite consistent. However, this doesn’t preclude having part of the theory stated outside of the context of the implementation. It merely says that all of the implementation is part of it. It also doesn’t obviate the development of a meta-theory; that is, an understanding of what parts of the theory are accurate, trustworthy, and significant. Such a meta-theory can be quite important in generating acceptance for the theory.

However, in general, Newell felt that a theory is adopted by an individual or community if it is useful to them. If it is not useful—because it is wrong in some way that matters to them, or because they do not trust or understand it, or because there is some other theory that is more attractive—then they will abandon it (or never even pick it up). He did believe that there should be a minimal threshold a theory should meet before being taken seriously, and that the threshold should evolve over time as the standards set by existing theories climb. But among the contenders, the ultimate success of a particular theory was more to be determined by how far you could get with the theory than by careful comparisons between it and other theories (though, of course, he did pay careful attention to a broad range of other theories, and particularly to whether they offered leverage in expanding the scope of extant unified theories). In the process, unsuccessful theories would just naturally be left by the wayside. He was thus much more concerned with extending his theories to cover new data than in detailed comparisons with other theories. In contrast, comparing his theories to data was always of central concern.

2.3.2. Identifiability

Fehling (along with Pollack) is concerned with the issue of theory identifiability, whether it ever really goes away, and whether it makes the modeling of behavior (rather than competence) impossible. It may be that in some fundamental manner the identifiability problem can never go away. How-

ever, what Newell probably meant here is not that the problem isn't still there in at least a technical sense, but that as the body of data gets larger, the issue of identifiability fades in significance because of the increasing constraints under which all of the theories must exist. If the total set of constraints on mind is finite, then ultimately any theory which satisfies them all is as good as any other. Even if there is an unbounded set of constraints—as Pollack might suggest—as long as the significance of the newly discovered constraints eventually follows a predominantly decreasing trend, the significance of the identifiability problem will also decrease correspondingly.

Even if you accept that the identifiability problem will always matter, the conclusion Fehling draws from this—that it is therefore only possible to focus on competence theories—just seems plain wrong. All scientific endeavors have identifiability problems, including the endeavor of developing competence theories—at best a competence theory provides one body of knowledge that is sufficient for generating some class of behaviors, but it does not show that it is the only such body of knowledge. What scientists in general must do is first find at least one theory that works, and then as other theories arrive, understand in what ways they are equivalent, and what the differences (if any) mean. In the area of unified theories of cognition we are still very much in the phase of trying to find just one theory that meets all of the relevant constraints. With respect to Pollack's concerns about basing a theory on a "Universal Programming Language", he is correct in surmising that Newell relies heavily on constraints derived from robust psychological results to distinguish among possible theories; however, these constraints are in no sense extra-disciplinary, as *UTC* is all about modeling the human mind.

2.3.3. *Form and content*

Pollack is concerned about the use of "natural law" versus "software law". However, it appears to us that he is conflating two distinct issues here—one of form and one of content—neither of which makes his point about *UTC*. The first issue is whether a theory is expressed in terms of mathematical equations versus process descriptions (i.e., what most programming languages provide). Neither form has an a priori claim to appropriateness as a basis for theory expression. What matters is whether a theory of the domain can be represented appropriately in the form—for example, whether a system's knowledge, goals, and processes can adequately be expressed—and whether the theory as so expressed supports answering the key questions we have about the domain. The second issue is whether or not a "theoretical" structure is *about* anything. Both mathematics and computer languages can be used to create arbitrary abstract structures that, while possibly quite

beautiful, may have nothing to do with how the world (or the mind works). When AI engages in such activities, it has much in common with pure mathematics (in both the positive and negative senses). However, when the structures are used to model natural activities—such as the phenomena people exhibit when behaving in the world (or in a lab)—what is being talked about is as much natural law as are theories expressed as equations. Even such rational-band theories as competence models are statements of natural law to the extent that they model, at some level of abstraction, classes of activities that do (or can) occur in the world.

Fehling is concerned that Newell is relegating all AI theories other than Soar to the ashbin of “content-free” frameworks. However, it looks to us like Fehling has misinterpreted two parts of Newell’s original statement. First, “content-free” is intended to refer to the fact that the frameworks make no content commitments that would keep them from being a neutral (but convenient) language for implementing any theory. It is *not* intended to imply that these frameworks are themselves vacuous in any way. Second, by saying that a unified theory of cognition is not a framework, Newell meant that it should really be a theory—that is, it should make commitments about how things work—rather than being a theoretically neutral (but expressive) framework or programming language within which anything could be encoded. Neither of these points was intended to denigrate existing frameworks, nor to say that other AI architectures could not be considered as possible unified theories of cognition.

2.3.4. *Competence theories*

Fehling believes that theories of competence are more useful than theories of behavior. However, utility clearly depends on application. If you actually do want to predict human behavior—so that you can teach effectively, or counterplan, or design an aircraft cockpit that minimizes the number of pilot errors, or any number of other important applications—a competence theory is not enough. Likewise, if you actually want to build a working system that can exhibit a behavior (or competence), a competence theory is not close to being enough. Being concerned with performance is clearly shared with the behaviorists—as in fact, it is shared with nearly all other approaches in cognitive psychology, AI, and cognitive science—but this certainly does not make the approach behaviorist. The behaviorists did get some things right.

2.4. *Natural language*

Dennett is concerned about the role natural language plays in providing a representational medium, and the extent to which this distinguishes humans from other animals. We’ll have to be quite speculative here, as we don’t know what Newell thought about this, and we are not ourselves experts in this area.

So, what we will do is use Soar as a stand-in for Newell. The most natural prediction from Soar here is that the key difference between human and other animal cognition is in the architecture, and that natural language is based on the basic efflorescence of adaptation provided by the human architecture, rather than itself being the primary source (see, for example, [9] for thoughts on natural language in Soar). However, this prediction doesn't rule out there being specialized problem spaces (and representations within these spaces) for natural language, and their providing general capabilities that can be recruited for use in arbitrary tasks. For example, recent work on *Linguistic Task Operators* in Soar is examining how natural-language representations, along with the operations defined on these representations, can be used in solving parts (or all) of specific problems [10].

Minsky wonders about the origins of nouns and verbs in Soar. This issue can also be dealt with only in a rather speculative fashion. However, it should not be too much of a stretch to imagine them as arising as the linguistic correlates of states (or parts of states) and operators, respectively. These are Soar's primitive notions of objects and actions.

2.5. Evolution

Understanding the evolutionary path by which the human mind evolved is a fascinating scientific problem, and one in which Pollack, Arbib, and Minsky are all deeply interested. Newell clearly understood the importance of the evolutionary constraint on the architecture—as evidenced by its presence in Fig. 1-7 (p. 19) as one of the core constraints on the shape of the mind—but was also just as clearly not ready to deal with it. We are also not ready to deal with this constraint in any detail, other than to notice, with respect to Minsky's comments, that evolution is constrained by both the material it has to start with—that is, the initial organism—and the environment in which the organism must exist (that is, the task environment). Only by understanding the interactions among these two constraints will we be able to get a true sense of the extent to which evolution prefers bizarre over elegant solutions and complex over simple solutions.

Despite the admitted importance of the evolutionary constraint, no one should be misled into believing that the ordering evolution imposes on cognitive functioning need provide the best—or even a particularly good—ordering in which to go about studying and modeling the mind. To use a problem-space metaphor, this would be equivalent to claiming that the best way to understand a state in a space is to understand its parts in the order they were generated by the sequence of operators that led to the state. What this neglects is that states (and organisms) tend to be structures with stable interactions among their parts—if organisms were not, they probably could not survive—and which can often best be understood directly in terms of

these interactions, rather than through the possibly convoluted means by which the stability was reached.

3. Soar

The following discussion of Soar is organized around whether the comments being responded to are about memory, scaling up (including integration), task domains, seriality, emotion, or comparisons with other architectures.

3.1. Memory

Schank and Jona are concerned about a lack of organization of Soar's productions into higher-level units, a lack of attention to how memory should be accessed, and a lack of emphasis on preparation (that is, use of memory) rather than deliberation (that is, search). Minsky is also concerned about memory access and about whether Soar can have specialized representations (and whether they can be learned, or must be innate).

3.1.1. Organization

We surmise that Schank and Jona are being misled by the surface syntax of the memory structures and are thus missing the deeper semantic organization that does exist. Soar's productions do not simply comprise a flat memory structure. Instead, they are semantically organized around the objects—i.e., the goals, problem spaces, states, operators, and other miscellaneous subobjects—to which they are relevant. For example, each problem space has a cluster of productions that define it. Likewise, each operator has a cluster of productions that define how it should be applied (as well as when it should be considered and when it should be selected). Such clusters of productions are only eligible for execution when their corresponding objects are active in working memory. When examining Soar's productions it is easy to miss this structure, as it is buried in the conditions of the productions. However, one of the main advantages of this approach is that a large variety of objects can be dynamically (and flexibly) constructed during execution—according to the cluster of productions whose conditions make them relevant—rather than being limited to only instantiations of those object classes predefined in something like a long-term frame hierarchy.

In addition to the organization provided simply by having these objects, we have recently come to realize that the objects themselves are organized into a full-blown new processing level, called the *problem space computational model* [14]. At this level, Soar is appropriately described as consisting of a set of interacting problem spaces, where each problem space effectively

realizes a constrained micro-world. Languages defined at the problem-space level, such as TAQL [21], specify systems directly in terms of problem-space components (and can be compiled automatically into Soar productions).

3.1.2. Specialized representations and innateness

Minsky is concerned about whether Soar has specialized representations. Soar definitely can utilize specialized representations because individual problem spaces may have their own representations that are constructed in terms of the uniform low-level attribute–value structure. This has proven adequate for the tasks so far investigated, but we do not yet know whether it will remain so in the future, or whether we will need to add new representations—for example, for spatial information—to the architecture.

Minsky also wonders about the knowledge-genesis issue of whether Soar would be able to acquire specialized representations, along with the ability to convert rapidly among them, or whether it really needs to be born with them. Soar currently doesn't come born with specialized representations—except for the limited way in which they are provided by the “default” rules that are always loaded into Soar on start-up—and though we don't currently know whether or not they could be learned in general, a two-pronged approach to resolving this issue does seem clear. One prong is to investigate in some detail whether (or how) such knowledge could indeed be learned by Soar. The other prong is to continue accumulating evidence about the capabilities humans have at birth. If the conclusion is that the representation and conversion capabilities must all be there at birth, there would then still be the question as to whether it exists as innate content (that is, as productions or whole problem spaces), or as architecture. Either could appear physically in the brain as “specialized sub-organs”.

3.1.3. Access

Schank and Jona are concerned that the issue of how memory should be accessed has received insufficient attention in Soar. However, the problem really appears to be not that it hasn't been dealt with in Soar, but that it has been dealt with in a different manner than in, for example, case-based reasoning. In case-based reasoning, the usual assumption is that memory consists of large-grained declarative structures that are processed little at storage time, except for some work on indexing. The understanding and adaptation of the case happen later, at retrieval time. In Soar, much more of the processing occurs at storage time. In fact, storage effectively occurs as a side-effect of the understanding process, resulting in the information usually being proceduralized with respect to the current context into a set of small-grain rules, each of which becomes indexed by those aspects of the situation that determined its consequences. The only form of auto-

matic (that is, architecturally driven) adaptation that occurs at retrieval time is instantiation and combination (of object fragments from multiple rules). More intensive adaptation of retrieved structures is possible, but must be driven by knowledge (i.e., productions) or by more deliberative processing in further problem spaces. It would be quite easy to get into an extended debate about the relative merits of these two approaches to memory access—and of whether phenomena such as *encoding specificity* (p. 321) favor one approach over the other—however, the important message to be taken from here is that significant attention has been paid to the indexing of knowledge in Soar. In fact, indexing is most of what the learning mechanism spends its time doing (i.e., figuring out what conditions to place on a production).

Minsky expresses related concerns about Soar having too easy a time in accessing arbitrary pieces of knowledge: "... in Soar all knowledge is assumed to be uniformly accessible, so there is nothing to bar the central processor from accessing anything ...". However, two factors make such accessing more difficult than it appeared to Minsky. The first factor is that, though the working-memory contents of one problem space can be examined by other problem spaces, the productions cannot be. Productions can only be executed, and then only in the context in which they match. The second factor is that the specialized representations which get built up within individual problem spaces can make it impossible for one problem space to understand the working-memory contents of another space without solving the kinds of integration problems that are discussed in Section 3.2.

3.1.4. *Preparation versus deliberation*

Here we can only surmise that our propensity to talk about how flexible Soar is as a searcher has misled Schank and Jona into believing that Soar favors deliberation (search) over preparation (use of memory). It was hopefully clear from *UTC* that Soar uses memory (encoded as productions) whenever it has it—and we usually try to make sure it does have it, or can learn it. Search only arises as a response to uncertainty and as a court of last resort when the available knowledge is insufficient (pp. 96–97). One of the things we have learned over the years is that problem spaces are about much more than search—they also provide a deferred-commitment control strategy that allows decisions to be postponed until run time (and which also turns out to make room for the kind of "dynamic planning" that Arbib is concerned about), constrained microworlds for focused deliberation, a forum for error recovery, and a means of bootstrapping into task formulation [14]. We thus have no strong quarrel with the claim that cognition is 90% preparation and 10% deliberation, though the exact numbers must be confirmed with good data.

3.2. *Scaling up*

Schank and Jona, Hayes-Roth, and Pollack are all concerned about scaling up, which is, of course, a major issue for any proposed intelligent system. Three subissues are raised by Schank and Jona:

- (1) *Size*: Whether the architecture can cope with large numbers of productions.
- (2) *Integration*: Whether you can scale the knowledge just by adding in the productions from a bunch of independent tasks.
- (3) *Transfer*: The amount of cross-contextual transfer of learning that will occur.

Pollack's concerns about the construction of large-scale systems and Hayes-Roth's programmatic suggestions both fit in well with Schank and Jona's concern about integration, and are therefore discussed in that context.

3.2.1. *Size*

Recent evidence shows that, for at least one significant task—message dispatching—Soar can scale up to over 10,000 productions without a significant loss of speed [3], and more recent results extend this to over 100,000 productions [2]. More clearly needs to be done here—and will be—however, these results should at least allay some of the scale-up fears.

3.2.2. *Integration*

None of us believe that you can build an integrated large-scale system by just adding together a bunch of productions created for different tasks. Because problem spaces provide Soar with an approximation to a modular structure, you often can just throw the problem spaces from multiple tasks together. However, what you get then is a system which can do the tasks individually, but not particularly a system that understands how to combine them in any effective way. There is integration that needs to go on with knowledge—i.e., among the problem spaces—just as there is for the architecture.

What we do know about knowledge integration in Soar is that problem spaces interact through impasses; that is, when a problem space gets stuck, additional problem spaces can be brought in to discover the information that will allow the original one to proceed. Chunking then provides a permanent transfer of the knowledge integrated together from the additional spaces to the original space. The integration of the chunked rules into their new space is relatively straightforward, as the nature of chunking automatically ensures that the new rules are already in the language of the space for which they are learned—the chunks' conditions and actions are all derived from existing working-memory elements in this space—and the decision cycle provides

an openness of knowledge access that enables new rules to contribute their preferences to whichever decisions they are relevant.

Problem-space interactions are normally modulated by hand-coded productions that:

- (1) determine which spaces are considered and selected for which impasses;
- (2) create initial and desired states in the new problem spaces as a function of the structure and content of the existing context hierarchy (particularly including the immediately preceding impasse and problem space); and
- (3) return results to the original problem space.

This approach has worked fine as long as it is possible to predefine the classes of interactions that will arise. However, it falls short of the ideal, which would be to provide the potential for any problem-space to be used for any impasse for which it might be relevant—and to thus enable knowledge from arbitrary spaces to migrate, via chunking, into other spaces as appropriate. Soar does provide a framework that may make this ultimate integration possible, in that it allows the activities of problem-space generation and selection, initial and desired state creation, and result returning to become first-class tasks for the system to work on. However, the super-analogical problem of how, in general, two previously-unrelated problem spaces can be brought into correspondence, so that one can aid the other, is a question that we are only now beginning to seriously address within Soar. It is a great problem though (and clearly not just for us).

Hayes-Roth recommends a program of research that is focused on the cumulation and integration of knowledge (probably from much the same intuition that underlies the CYC project). That seems like a fine idea to us, though clearly a very difficult enterprise, as Hayes-Roth acknowledges. In our own judgement we have so far been reasonably successful in following her recommendation with respect to the architecture, but much less so with respect to the content within the architecture. The reasons are several, including:

- (1) the architecture is tightly controlled by a small group of people, whereas the content is developed diffusely over a large loosely-coupled distributed community; and
- (2) the architecture contains a relatively small number of reasonably robust algorithms, whereas the content houses a large amount of fragmentary, ill-specified, heuristic knowledge (as it should).

The research effort mentioned above, as well as some of the new tasks we are going after (partly described in Section 3.3), are intended to help drive

us further in the direction of knowledge integration. Even so, it can still frequently be useful to back off from the ideal Hayes-Roth presents, and to look at independent integrations of partial capabilities in specific tasks. When it is not possible to see a straight-line path to the integration of a complete new robust capability—which is, in fact, frequently the case—such partial integrations can be a great way to build the scientific base that should eventually support the full integration required (and can also be an easy way to incorporate many loosely coupled researchers).

Pollack proposes “building systems from the bottom up using robust and stable cooperatives of goal-driven modules locked into long-term prisoner’s dilemmas” (presumably based on nonlinear dynamical systems). There is a lot of general common sense in this prescription—creating stable, robust modules and developing a stable organization of them—though the effectiveness of the particular approaches he advocates (such as long-term prisoner’s dilemmas) is much less clear, as is the relationship of these approaches to existing large-scale system-construction methodologies that attempt to resolve the same basic issues. Specifically with respect to Soar, we see that problem spaces provide “goal-driven modules” that interact through the relationship of impasse resolution. Within problem spaces there are semi-independent objects (states, operators, etc.) each constructed out of productions (or more problem spaces). The productions themselves tend to be significantly more independent than those in most systems because of the lack of conflict resolution—productions fire in parallel, with behavioral arbitration occurring outside of the production system, via preferences and the decision procedure. All of this gives Soar a form of modularity that should help in the construction of large-scale systems. However, it is impossible to tell at this time whether this is enough—even when combined with the growth potential provided by chunking—or whether it is any better or worse than the approach Pollack describes.

3.2.3. *Transfer*

We do currently see a lot of transfer within and across problems from the same class of tasks; however, we do not yet see a great amount of cross-contextual transfer. Since such transfer is likely to be a direct function of the amount of commonality among the processing across different contexts, it is also probably related to the degree to which particular problem spaces can be used in different contexts, and thus a function of the amount of integration that can be achieved (as just discussed). Additional integration will thus probably help, but we would never expect to see a huge amount of cross-contextual transfer. The evidence on analogy and encoding specificity suggest that such transfer is actually quite limited in humans.

3.3. Task domains

Both Hayes-Roth and Schank and Jona are concerned about the set of tasks used in driving the development of Soar. Hayes-Roth is concerned that the psychological tasks are too low-level and artificial to be informative—and may just “reflect artifacts of biological hardware”—and that the AI tasks may have nothing to do with intelligent behavior (or might reflect super-intelligent behavior). Schank and Jona are concerned that paying attention to “unrealistic and unnatural” psychological tasks—as opposed to the more realistic tasks that AI can study but that psychology has more trouble studying—will both lead to a distortion in the resulting theory and distract effort away from the more important realistic tasks. Let’s divide this issue up into two parts: (1) whether there is a right set of tasks to be working on; and (2) if so, what they are, and where Soar stands with respect to them.

On the first part, we are pretty sure we know what Newell would have said: There is no right set of tasks. People are not special-purpose devices that can only work on one type of task. Indeed, any single task can lead to a micro-theory that is overoptimized with respect to its own narrow domain, and that is thus severely distorted with respect to the full complexity and richness of human cognition. So, what really matters is not what task is covered, but what diversity of tasks are covered.

On the second part, we sympathize with the call for AI to study more realistic and naturalistic tasks. A key feature of such tasks is that they often contain within themselves enough essential diversity to keep the resulting micro-theory from being extremely narrow to begin with. The Soar community has put a fair amount of energy into understanding how Soar can perform in reasonably complex knowledge-intensive domains, both in terms of replicating existing systems and in terms of constructing novel ones. The domains range over such areas as computer configuration, medical diagnosis, algorithm design, blood banking [7], message dispatching [3], factory scheduling [5], browsing [15], and playing video games [6]. These are realistic tasks that people do perform, but they may seem to fall somewhat short on the scale of “naturalism”. We have not yet looked much at performing everyday tasks, and that would be a useful addendum to the classes of tasks that we have examined. Perhaps the closest we have come is nascent work on how Soar can be used as the basis for developing human-like intelligent agents in large-scale, simulated, physical environments. This is a “real” task in the sense that the simulation environments are developed by groups outside of the Soar community, the environments have significant value for those groups, and they have a direct need for intelligent agents in these environments. This is also a reasonably naturalistic task, as it deals with multiple intelligent agents—some human-controlled and some

computer-controlled—in realistic physical situations; however, they do not quite tend to be “everyday” situations.

Despite our general sympathy with the reviewers’ calls to more realistic and naturalistic tasks, we also believe that controlled experimentation with more limited tasks can provide valuable information that can be difficult-to-impossible to derive from more complex tasks; for example, the rich bodies of data on controlled versus automatized performance, on memory, and on practice. Newell has already stated the rationale for this better than we could have, but the basic idea is that these short time-scale tasks—i.e., tasks requiring hundreds of milliseconds up to small numbers of seconds—are the ones that get close to the architecture, by eliminating much of the flexibility that humans have at longer time scales. To date, Hayes-Roth’s concern about these phenomena being mostly artifacts of the biology (or in Newell’s terms, phenomena of the biological band) have not been realized. We have been successful in modeling a number of tasks at this level—such as visual attention [20] and transcription typing—with functional models based on $\sim\sim 50$ msec operator-execution times. In general, as long as you don’t restrict yourself to the micro-theory developed from a single such task, and in fact strive for coverage across a broad range of both realistic and controlled tasks, the danger of getting a fundamentally distorted view can be reduced greatly, while the amount of useful information available can be increased greatly.

3.4. Seriality

Minsky is concerned about Soar’s “selecting and executing productions—that is, IF-THEN rules—one at a time”. However, here Minsky seems to have fundamentally misunderstood the parallel nature of Soar. It is serial at the level of problem-space operations, but quite parallel below that: productions match and execute in parallel, and preferences are generated and decisions made in parallel. (Of course, when we implement Soar on a serial machine, this parallelism can only be simulated, but parallel implementations have also been investigated [19]. Because Soar does embody this parallelism, many of Minsky’s arguments about what would be difficult to do quickly enough in Soar don’t go through. To take just one specific example, recent work on garden-path sentences in Soar shows precisely the kind of rapid shift of interpretation about which Minsky is concerned. In this work, an ambiguous sentence is not a garden-path sentence precisely when it is possible to repair, in real time, from the incorrect interpretation of the sentence to the correct one [11].

Arbib is also concerned about the seriality of computers (and Soar) versus the parallel/distributed nature of brains. Part of the problem with this concern is that, as with Minsky, Arbib appears to have missed Soar’s inherently

parallel nature. However, a further problem with this claim is that there is strong evidence for something like serial processing in human cognition; for example, the large body of results on (parallel) automatized processes versus (serial) controlled/attentive processes [18]. In his final remarks, Shiffrin summarizes as follows: "Behavior in general is accomplished by limited, perhaps serial, attentive processes operating in parallel with numerous automatic processes, with the two systems passing information back and forth at all levels of analysis." (p. 805). Soar is, at least at a gross level, consistent with these results, as the productions produce (parallel) automatized behavior while the problem spaces produce (serial) controlled behavior.

3.5. *Emotion*

Dennett's primary concern about Soar is its being all business, with no pain, pleasure, playfulness, laziness, etc. There is no question that this is a rather large gap with respect to Soar's coverage of human cognition. Even if you abstract away from the purely biological aspects of emotion, there are large and well-documented effects of emotion on cognition—in many ways the issues are not unlike those concerning the interactions between cognition and perceptual-motor systems. So it is an important constraint, but not one that we yet know how to deal with. It will clearly need to be addressed some day and, when it is, it will almost certainly have an impact on the architecture.

3.6. *Comparisons with other architectures*

One of the things Minsky does in his review is to compare Soar directly with the Society of Mind. On the whole, we think he did an excellent job. The comparison helped us both to understand the Society of Mind better, and to understand some of the ways in which it does map quite nicely onto Soar (and vice versa). However, there are two aspects of the comparison to which we would like to add.

The first addition is that, rather than mapping the Society of Mind's management function onto Soar's problem spaces, we would map it onto Soar's decision procedure, along with the knowledge and processing that generates the preferences used by the decision procedure. This knowledge may consist simply of productions that directly generate preferences upon execution, or it may consist of problem spaces that require arbitrary amounts of processing before being able to generate the appropriate preferences. When management is by the former, it doesn't require impasses or deliberate processing, and can proceed in parallel for many activities. This would also be quite compatible with Minsky's notion that "The managers themselves are installed as consequences of impasse-resolving episodes", as these are

exactly the circumstances under which chunking installs new productions in Soar.

The second addition concerns Soar's apparent lack of the concept of a "difference", which is so central to the Society of Mind. Soar does indeed not have a large-grained, architectural, difference-detection mechanism that compares arbitrary pairs of structures. However, it does have at its very core a fairly general match capability that provides a small-grained difference-detection mechanism; that is, it can detect situations where a value bound in one condition of a production is not the same as the value bound in another condition. This capability supports, but does not mandate, large-scale structure comparisons. An important part of the philosophy underlying this choice is that difference detection is not a purely syntactic operation that can be applied to two arbitrary structures and yield meaningful answers. Instead it must be a knowledge-guided process that can be based on which differences really matter in particular contexts. The selective use of difference detection in the matcher is one way of accomplishing this. This choice is also consistent with the overall philosophy underlying Soar's approach to problem solving. In contrast to systems such as GPS, Soar is not locked into a single problem solving method such as means-ends analysis (MEA). Instead it is to be free to use whatever methods are supported by the knowledge it has about the task. Thus, if it has difference information, something like MEA is possible; when it doesn't have such knowledge, other methods should be appropriate; and when it has such knowledge plus additional knowledge, more powerful, possibly hybrid, methods should be usable.

Arbib also compares schema theory with Soar, and generates similar conclusions. In particular, he proposes to map schemata onto problem spaces, but is then concerned about the resulting seriality. However, as with the Society of Mind's management function, it is not at all clear why the mapping shouldn't be extended to, at least, include Soar's productions as schemata.

4. Summary

If we pull up from all of the details, there are three major themes that we hope the reader takes away from this response. The first theme is that studying the architecture is important. No matter how much domain-specific power comes from the knowledge, there is no way to build a unified and effective theory of cognition without significant study of, and effort on, the architecture.

The second theme is that no band of phenomena is special. Each band has its own data, methods, regularities, and applications. A complete model of human behavior requires understanding all of them. Newell focused

primarily on the cognitive band in his career, and *UTC* represents his attempt to build a comprehensive theory of the architecture—that is, the fixed structure—that gives this band its distinct shape. Unified theories of the other bands may take on quite different shapes, as might unified theories of cognition that are based on other phenomena within the cognitive band.

The third theme is that there is plenty of room—in fact, Newell tried to explicitly encourage it—for others to develop their own unified theories of the cognitive (or any other) band. As the theories grow, and the set of phenomena covered by them begins to converge, we expect that some of the theories will die (from an inability to grow further), while others will themselves start to converge, if not in their surface structure, then at least in their essence. Be warned though, that the development of such theories can be an immense undertaking, requiring the integration of a rather vast amount of both expertise and manpower. And even then, there will always be significant gaps and inconsistencies that can be criticized. Nonetheless, there are few research paths more exhilarating than trying to put it all together.

References

- [1] B. Cho, P.S. Rosenbloom and C.P. Dolan, Neuro-Soar: a neural-network architecture for goal-oriented behavior, in: *Proceedings Thirteenth Annual Conference of the Cognitive Science Society*, Chicago, IL (1991).
- [2] B. Doorenbos, *Personal communication* (1992).
- [3] B. Doorenbos, M. Tambe and A. Newell, Learning 10,000 chunks: what's it like out there? in: *Proceedings AAAI-92*, San Jose, CA (1992).
- [4] W.D. Gray, B.E. John and M.E. Atwood, Project Ernestine: validating GOMS for predicting and explaining real-world task performance, NYNEX Science and Technology Center and School of Computer Science, Carnegie Mellon University, Pittsburgh, PA (1991).
- [5] W. Hsu, M. Prietula and D. Steier, Merl-Soar: scheduling within a general architecture for intelligence, in: *Proceedings Third International Conference on Expert Systems and the Leading Edge in Production and Operations Management* (1989).
- [6] B.E. John, A.H. Vera and A. Newell, Towards real-time GOMS, Tech. Rept. CMU-CS-90-195, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA (1990).
- [7] T.R. Johnson, J.W. Smith, K. Johnson, N. Amra and M. DeJongh, Diagrammatic reasoning of tabular data, Tech. Rept. OSU-LKBMS-92-101, Laboratory for Knowledge-based Medical Systems, Ohio State University, Columbus, OH (1992).
- [8] J.E. Laird, M. Hucka, S.B. Huffman and P.S. Rosenbloom, An analysis of Soar as an integrated architecture, *SIGART Bull.* 2 (1991) 98–103.
- [9] J.F. Lehman, R.L. Lewis and A. Newell, Integrating knowledge sources in language comprehension, in: *Proceedings Thirteenth Annual Conference of the Cognitive Science Society*, Chicago, IL (1991).
- [10] J.F. Lehman, A. Newell, T.A. Polk and R.L. Lewis, The role of language in cognition, in: G. Harman, ed., *Conceptions of the Human Mind* (Erlbaum, Hillsdale, NJ, to appear).
- [11] R.L. Lewis, Recent developments in the NL-Soar garden path theory, Tech. Rept. CMU-CS-92-141, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA (1992).

- [12] A. Newell, SOAR as a unified theory of cognition: issues and explanations, *Behav. Brain Sci.* **15** (1992) 464–488.
- [13] A. Newell, Unified theories of cognition and the role of Soar, in: J.A. Michon and A. Akyürek, eds., *Soar: A Cognitive Architecture in Perspective* (Kluwer Academic Publishers, Dordrecht, Netherlands, 1992).
- [14] A. Newell, G.R. Yost, J.E. Laird, P.S. Rosenbloom and E. Altmann, Formulating the problem space computational model, in: R.F. Rashid, ed., *CMU Computer Science: A 25th Anniversary Commemorative* (ACM Press/Addison-Wesley, New York, 1991).
- [15] V.A. Peck and B.E. John, Browser-Soar: A computational model of a highly interactive task, in: *Proceedings CHI'92* (ACM Press, New York, 1992).
- [16] P.S. Rosenbloom, J.E. Laird, A. Newell and R. McCarl, A preliminary analysis of the Soar architecture as a basis for general intelligence, *Artif. Intell.* **47** 289–325 (1991).
- [17] B. Scharf and A.J.M. Houtsma, Audition II: loudness, pitch, localization, aural distortion, pathology, in: K.R. Boff, L. Kaufman, and J.P. Thomas, eds., *Handbook of Perception and Human Performance*, Vol. I: *Sensory Processes and Perception* (Wiley, New York, 1986).
- [18] R.M. Shiffrin, Attention, in: R.C. Atkinson, R.J. Herrnstein, G. Lindzey and R.D. Luce, eds., *Steven's Handbook of Experimental Psychology*, Vol. 2: *Learning and Cognition* (Wiley, New York, 2nd ed., 1988).
- [19] M. Tambe, D. Kalp, A. Gupta, C.L. Forgy, B. Milnes and A. Newell, Soar/PSM-E: investigating match parallelism in a learning production system, in: *Proceedings ACM/SIGPLAN Symposium on Parallel Programming: Experience with Applications, Languages, and Systems* (1988).
- [20] M. Wiesmeyer and J.E. Laird, A computer model of 2D visual attention, in: *Proceedings Twelfth Annual Conference of the Cognitive Science Society*, Cambridge, MA (1990).
- [21] G.R. Yost and E. Altmann, TAQL 3.1.3: Soar Task Acquisition Language user manual, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA (1991).