J. Expt. Theor. Artif. Intell. 1(1989)249–253.

Response 1

Commentary on Jon Sticklen's 'Problem-solving architecture at the knowledge level'

WILLIAM J. CLANCEY

Institute for Research on Learning, 2550 Hanover Street, Palo Alto, CA 94304, USA

Sticklen's ambitious paper seeks to relate Newell's 'knowledge level' (KL) description of cognitive systems to engineering principles for constructing expert systems. Unfortunately, the analysis suffers throughout by confounding psychological studies of human problem-solving with engineering principles for designing computer programs. The author appears to want to make a contribution to both areas, but is never clear on whether scientific (study of intelligence) or engineering (expert systems) standards are to be applied to his 'architecture hypothesis'. Most importantly, Sticklen appears to have missed the whole point of what kind of theory the KL hypothesis is and in what respect it is a hypothesis.

From the start, in Sticklen's discussion of the 'test of theories stated at the knowledge level', it is never clear what phenomena are being studied. For a theory to be tested scientifically, it must clearly address some natural phenomenon. Human behavior is apparently the domain Sticklen wishes to address, but without any explanation at all, the text jumps abruptly to engineering: 'our sense of "prescription" is that the theory should offer guidance in the construction of problem-solving systems. . . .'. Later he says, 'Schools of thought such as the GT approach to problem-solving propose that the roots of cognition. . .', revealing again the implicit mixing of an engineering enterprise (designing computer representations of processes) with psychology (explaining the roots of cognition in people). The generic task research of Chandrasekaran is surely valuable for designing tools for expert systems. Possibly a better engineering methodology will yield useful psychological models. However, attempting to do both of these at once without clearly adhering to corresponding scientific and engineering methods of evaluation might doom the effort on all accounts. This danger is most clear in Sticklen's emphasis on predictive power, which always gets translated in his analysis to 'show me how to build a program', and not psychological experiments that make predictions about human behavior.

At the heart of Sticklen's dissatisfaction with the KL hypothesis is his belief that a scientific theory must have predictive power. In early stages many natural sciences produced only taxonomies of phenomena to be explained, medicine and biology being prominent examples. The KL hypothesis is a claim that cognitive systems can be *described* at an implementation-free level, that is, without making any claims for explaining how they work. Similarly, early biologists classified bacteria according to the environments they grow in, stimuli they respond to, etc. – without making commitments about the as yet unobserved mechanisms. From this perspective, the KL hypothesis is that we can model cognitive systems by comparing knowledge capacity, characterizing potential changes in knowledge capacity due to learning and environmental effects, and so on.

0952–813X/89 $3.00 Taylor & Francis Ltd

A quite similar perspective is used in linguistics, as Newell explicitly notes in his paper when he calls the KL hypothesis a 'competence theory'. Natural language grammars needn't correspond to mental structures; rather they are just our descriptions of regularities in sentences that we perceive as observers. Saying what is in the head that produces sentences – the mechanism – is a different matter. Thus, it is not so much that the KL hypothesis of Newell is incomplete; rather that it is a claim that useful theories of behavior can be descriptive, not explanatory in the mechanistic sense of specifying structure and function of the organism/machine that produces such behavior.

Indeed, the very hypothesis to be tested is that knowledge can be characterized independently of a structural level at all! Sticklen appears not to understand this point in saying that if the KL exists, it must have an architecture. The KL is inherently a behavioral characterization. By contrast, the memory-cpu level of a computer does exist physically, and it is at a level higher than transistors and resistors, etc. The KL is a certain kind of specification that *by definition (by hypothesis)* says nothing about structural/functional decomposition or agents or messages, etc.

In fact, there is evidence that the knowledge-level description, albeit idealized like natural language grammars, does offer predictive value. To see this, observe that as a certain *type* of analysis, KL analysis must be applied to a particular cognitive system before it can offer predictive value. For example, consider Dietterich's KL analyses of particular learning programs (Dietterich, 1986). He shows that a KL description of a learing program enables us to predict the space of concepts a particular program will be capable of learning – independent of the program's specific architecture. Another example is provided by Alexander's 'KL analysis' (Alexander *et al.* 1986), which demonstrates how a KL description can be used to specify the space of concepts and operations an expert system will be capable of performing. This specification can then be instantiated in any number of possible computer architectures. The power of the KL hypothesis is that it shifts our attention from worrying about how to build a system, to characterizing what behaviors must be possible. Contrast this with Sticklen's constant frustration that the KL hypothesis 'gives us no clue about how to start building. . .' The whole idea is that it tells us to start *by not talking in terms of architecture*!

As a special case the KL hypothesis is proven for computers by theorems that show that every computer that is Turing equivalent is a universal computer, that is, it is capable of computing the same functions. Empirically, we have demonstrated over the past few decades that there are many computer architectures that are Turing equivalent. The theory of computation, backed up by engineering realization of alternative architectures, provides at least theoretical support to Newell's belief that the capacities of cognitive systems found in nature, such as the human mind, can be described usefully at a non-structural level. Newell's hypothesis suggests that we seek the equivalent of 'Turing equivalence' computability theorems for characterizing intelligence. While Turing's work predated the development of modern computers, it provided a theoretical basis that obviated fruitless searches for alternative, more powerful computing machines.

We can argue about a few other points Sticklen raises, some of which are quite important.

First, in my original analysis of MYCIN, I claimed that the inference structure (e.g., as shown in figure 3 of Sticklen's paper) is independent of the order of which inferences are made (the inference process). My intent was to show graphically that we could characterize what inferences a cognitive system is capable of making, independently of how the knowledge is encoded and how it is applied to the problem at hand. In adding inferences between figures 2 and 3, Sticklen claims that the analysis 'gives no guidance on which inference we should *expect* in a given problem solving situation'. But this is like saying that natural language grammars are useless because they don't enable us to predict what someone is going to say! The whole idea is that we can find regularities in the utterances after they are made: as psychologists we can grammatically model a problem-solver's capabilities, analyzing a protocol at this level; as designers we can specify a program's capabilities in terms of the types of utterances it must be capable of making. KL descriptions can also be used, like natural language grammars, to predict whether members of a specialist community will find a given problem solution to be sensible. By factoring out the inference process as well as what underlying representations (if any) are used, we can use the inference structure to compare solutions to a problem, thus identifying common and alternative lines of reasoning. We could show, for example, that one person used classification to solve a diagnosis problem, while another used causal reasoning at the level of internal states.

Sticklen's discussion of simulation is bizarre and revealing. He says, 'A simulation of a problem-solving agent would itself be a problem-solving agent.' What does this imply? Suppose it is 1889 and I have just constructed a hot air balloon that can soar into the air like a bird. I tell you, 'This is a simulation of flight.' What would be accomplished by saying, 'The hot air balloon is a kind of bird because it can fly?' We must avoid supposing that a phenomena can be reduced to the examples that illustrate it. Just as flight is much more than getting something into the air, intelligence is much more than what any of our programs have accomplished. Indeed, just as for the early flight engineers, we have almost certainly not even identified all of the phenomena we need to replicate. For example, until we identify a characteristic like 'lift', we don't even realize what characteristics of flight might be important for controlling flight. In saying that any given simulation of problem-solving is a problem-solving agent, we do gross injustice to the exceedingly complex and wide-ranging phenomena we seek to describe and replicate. We make it sound like one example, one program, captures the full range of capabilities that makes a human being a problem-solving agent. Thus, in one fell swoop Sticklen reduces human beings to 'problem-solving agents' and then to 'today's programs'.

Strikingly, from this perspective the well-known effect in AI of saying that what has been accomplished is not intelligence, is perfectly understandable – we are still trying to identify examples of intelligence and its essential characteristics. Hot air balloons were an important technological advance, but they are too slow and unreliable for crossing the Atlantic and totally useless for getting to the moon. It is important to point out that a hot air balloon is still not a bird, to remind ourselves that flight is much more than rising into the air and drifting with the currents. A chess-playing program is still not a human-like problem-solving agent, because intelligence is much more than optimizing moves in an axiomatic game. Each accomplishment defines some capability, but also allows better discrimination of the behavior we seek to replicate.

Sticklen's discussion of Marr's analysis hits on important points about mechanism versus behavioral descriptions. However, the distinction made in the statement 'not in the sense that it is a mechanism, but rather that it embodies a way to perform an important task' could be stated more crisply. Doesn't 'embodies' suggest a structure-function specification, and hence a mechanism? Again, the idea of different types of theories is not developed here. We must be clearer about the distinction between grammatical simulations (an apt description of most cognitive models developed to date) and theories whose structures and functions are intended to be mapped to physical processes in a human brain. In particular, we must be very careful when considering Marr's protein folding example. Marr's Type II theoretical analysis is meaningfully applied to protein folding because there are *physical constraints* that can be manipulated to determine the ultimate configuration of the molecule. Any process that produces a configuration design will have to adhere to these constraints. Thus, what every protein-folding mechanism must accomplish can be mathematically characterized, independently of how the configuration is computed. In what way is human problem-solving similar? The obvious analogy that human problem-solving is like solving protein puzzles glosses the complexities of human interaction, of how perception and representations arise dialectically in our conversations, and of how problems and theories are defined by articulating new concepts that objectify a social sense of reality (Berger & Luckmann 1967). It is in this respect that the KL hypothesis should be called into question – not that it fails to explain how intelligence is possible – but in its suggestion that knowledge is a property of individual people, rather than a capacity of a group of people forming a community, who mutually define what each individual perceives and seeks to accomplish.

At this deeper level, despite his repetition of Newell's points in section 4, Sticklen's discussion continues the AI/cognitive science view that knowledge is a substance that is stored in the problem-solving agent. This is evident in figure 4, which shows agents transmitting messages in some langauge, 'the vocabulary of knowledge organization and control'. It is precisely this kind of assumption that Newell aims to avoid in stating the KL hypothesis, viewing knowledge as a capacity to behave and not a collection of representations. Any discussion of knowledge organization and control misses the point. Newell's recent work, continued in his Harvard Lectures on 'Unified theories of cognition', (1987) is his first major attempt to break away from the idea that knowledge is stored in memory as representational structures. Sticklen's view of agents conversing inside the head continues the confusion about language, representation, and symbols that Newell calls into question.

In conclusion, I believe that we should view the generic task approach as a knowledge engineering methodology and not confuse it with discussions of intelligence. This will enable us to best communicate AI's qualitative modelling methodologies to scientists and engineers and develop useful tools for modeling complex processes. Those interested in the study of intelligent systems in the large, particularly human psychology, should realize that talking about a 'problem-solving architecture at the knowledge level' is like talking about pistons and fuel injection when somebody wants to know whether you can drive to New York.

I have been sharply critical of Sticklen's paper because I believe that clear, strong statements are important. In large part I believe AI has suffered from lack of commitment by researchers, including vague hopes about the relationships between computer programs and human beings, as well as unclear goals in producing useful computer tools vs. studying intelligence. I believe Sticklen's paper is useful precisely because it invites such a strong response. The connection between heuristic classification and Marr's analysis is quite apt (and was originally missed by me). The attempts to understand in what sense the KL hypothesis is a hypothesis and to reconsider architectural theories strike me as central concerns that need to be discussed. Everyone is struggling to understand these issues; the emerging consensus is that intelligence is both simpler and more subtle than our programs suggest. In effect, the above is an argument with myself, against my own ways of thinking, and a struggle to make clear to myself what I believe. So I would like to end on a positive note by thanking Sticklen for tackling a difficult problem head on and inviting discussion of his ideas.

References

Alexander, J. H., Freiling, M. J., Shulman, S. J., Staley, J. L., Rehfuss, S., and Messick, M. (1986) Knowledge level engineering: ontological analysis. *Proceedings of the National Conference on Artificial Intelligence* (Los Altos, CA: Morgan-Kaufrand), 963–968.

Berger, P. L. and Luckmann, T. (1967) *The Social Construction of Reality: A Treatise in the Sociology of Knowledge* (Garden City, NY: Anchor Books).

Dietterich, T. G. (1986) Learning at the knowledge level. *Machine Learning,* **1**(3), 287–316.

Newell, A. (1987) Unified theories of cognition. *The William James Lectures* (Cambridge, MA: Harvard University).