Functional principles and situated problem solving

William J. Clancey
Department of Computer Science, Knowledge Systems Laboratory, Stanford University, Palo Alto, Calif. 94304

Anderson's distinction between algorithm and implementation is useful, intuitive, and argued well. One could perhaps question his description of the algorithmic level in terms of actual "procedures that run in the mind," but a more concrete argument can be made against his claim that cognitive principles do not exist at the algorithmic level.

Anderson writes, "Only the implementational level can be understood in terms of general principles of cognition that are constant across different situations. The algorithms we possess are adapted to specific task demands and are as varied as those task demands." In contrast, recent expert systems research attempting to design "generic tools" tends to support Gleitman (1983): Recurrent knowledge organization and inference procedures for general tasks (e.g., diagnosis, planning, control) in different domains (e.g., medicine, electronics) can be abstracted from individual behavior. Indeed, these results are reflected in how expert systems researchers use the word "task" – a *kind of problem* (such as diagnosis or programming), not a specific problem to solve (patient to diagnose or program to write).

Anderson's analysis is apparently biased by research focusing on relatively formal problems, such as geometry and LISP programming. From the perspective of expert systems developed for scientific and engineering problems, cognitive science research in mathematics, typing, programming, and so forth is knowledge-impoverished. To capture what Anderson calls "a true functional level of the human mind," we must consider tasks that relate a person's behavior to some nonformal world. In general, both everyday and complex problem solving outside of formal domains like mathematics involve modeling the world in order to take action. Making selective observations, we construct and test alternative situation-specific models (e.g., alternative descriptions of disease processes in a particular patient), and relate them to action plans (e.g., therapy plans). To understand "how our cognitive mechanisms adapt to functionally important problems," as Anderson says, we must look at problems in which the problem solver is *situated;* that is, we must study problems in which the problem solver is faced with constructing a model of the outside world, within some social setting, and relating it to the needs of some task.

More specifically, engineering problems studied in expert systems research involve modeling some system in the world (a device, a manufacturing plant, a human body, a circuit, etc.) that the person is trying to design, repair, assemble, identify, diagnose, control, and so forth (called "generic tasks" [Chandrasekaran 1984; Clancey 1985]). Engineering problem solving of this type involves a modeling step to describe the world and a planning step to choose a course of action. Anderson's LISP and geometry problems involve no world to interpret; they have no functional significance in themselves. Rather, they are formal modeling tools that would be used in the context of some larger system-manipulating task, involving goals for doing something with this system in the world. This antecedent, mostly qualitative problem solving, which expert systems research focuses upon, provides the analysis that leads to a theorem to prove, a program to write, or an equation to solve.

Qualitative models can be described on different levels: the task and system, the computational method (classification vs. construction), the relational network representation (e.g., prototype hierarchy, state-transition graph, procedural hierarchy), and the implementation in a program (rules, frames, objects, etc.) (Clancey 1986). Recurrence or "principles" include both a vocabulary of relations for abstracting *processes in the world* (e.g., cause, progression over time, severity, location, flow-volume characteristics) and *cognitive processes* (often called "inference procedures") for describing complex processes in the world by explaining and predicting their behavior. For example, routine diagnostic problem solving in medicine and sand-casting can be modeled by a common set of knowledge structures and inference procedure (Thompson & Clancey 1986).

Cognitive principles of this type are not necessarily explicitly stored in the brain or even articulatable by the problem solver. Rather, they are abstractions of a grammatical form that express commonalities in the behavior of individual problem solvers. These abstractions include both *kinds of patterns* experts can articulate (familiar problem-solving situations and familiar courses of action) and recurrent *changes in attention* and *rationales for making observations* when forming a model. An example of such recurrence is the process of "triggering" a partial model on the basis of a few observations. Triggering reflects both the cognitive ability to usefully relate a new situation to past experience and the properties of a world in which processes tend to recur. Thus, the study of recurrence of processes in cognition and the world are complementary, involving the interaction of task resources and demands.

Analysis at this level contradicts Anderson's remark that "one cannot use protocol data to analyze a skill that is already compiled." In complex problem solving such as medical diagnosis, we abstract sequences of data requests (observations made by the physician) by relating them to changes in the situation-specific model (Clancey 1984). Moreover, if the problem solver has a model of how he reasons, as some good teachers do, we can ask for his description of the functional modeling goals that lie behind his questions (e.g., to detect erroneous data, or to establish temporal boundaries on the underlying cause).

The heuristic classification (HC) model of problem solving was developed to describe expert systems, but it is also a hypothesis describing human problem solving. The HC model claims that expertise (knowledge based on experience) consists of the ability to recognize situations by abstracting specific observations and relating these systems models to abstract courses of action, which are subsequently refined to meet the needs of the specific situation. Theories of problem solving based on such a model of experiential knowledge describe a computational method (HC), the modeling requirements of a task (e.g., testing hypotheses, discriminating among alternative system models), and the world (e.g., nature of the recurrence in the domain, urgency, efficiency, cost of observations, importance of ordering observations). Strikingly, problem-solving research in geometry, LISP and *pascal* programming, subtraction, algebra, and so forth ignores the inherent difficulties of *modeling nonformal systems*, in which data are uncertain and incomplete, system functionality is not axiomatic, and no written calculus exists. Consequently, this research presents an impoverished view of experiential knowledge structures and inference.

In conclusion, Anderson's call to the algorithmic level is reasonable, but application tasks for functionally important problems must be "situated," if we are to capture cognitive principles at this level. By situated, we mean, first, that the task involves explaining and predicting events in the world in order to plan courses of action (which will in turn satisfy higher goals), and second, that the problem-solving activity is itself constrained by a social context. In expert systems research focusing on "generic tasks," cognitive principles at the algorithmic level include *representation requirements for modeling processes in the world* (i.e., what is articulatable from experience must bear a useful relation to the complexity of the world) and *inferential competence* (i.e., the constraints problem solvers for similar tasks in different domains must satisfy when gathering information and manipulating representations in the process of formulating adequate situation-specific models and action plans).