The Knowledge Level Reinterpreted: Modeling Socio-Technical Systems

William J. Clancey

Clancey, W. J. 1993. *International Journal of Intelligent Systems*, 8 (1) 33-49. Special Issue: Knowledge Acquisition as Modeling, K. M. Ford and J. M. Bradshaw, editors.

.

Abstract

Knowledge acquisition is a process of developing qualitative models of systems in the world—physical, social, technological—often for the first time, not extracting facts and rules that are already written down and filed away in an expert's mind. Models of reasoning describe how people *behave*—how they interactively gather evidence by looking and asking questions, represent a situation by saying and writing things, and plan to act in some environment. But such models are inherently brittle mechanisms: Human reinterpretation of rules and procedures is metaphorical, based on pre-linguistic perceptual categorization and non-deliberated sensory-motor coordination.

This view of people relative to computer models yields an alternative view of what tools can be and the tool design process. Knowledge engineers are called to participate with social scientists and workers in the co-design of the workplace and tools for enhancing worker creativity and response to unanticipated situations. The emphasis is on augmenting human capabilities as they interact with each other to construct new conceptualizations—facilitating conversations—not just automating routine behavior. Software development in the context of use maintains connection to non-technical, social factors such as ownership of ideas and authority to participate. The role of knowledge engineering is not merely "capturing knowledge" in a program delivered by technicians to users. Rather, we seek to develop tools that help people in a community, in their everyday practice of *creating* new understandings and capabilities, new forms of knowledge.

A Shift in Perspective

Knowledge acquisition is a process of developing computer models, often for the first time, not a process of extracting facts and rules that are already written down and filed away in an expert's mind. We can represent knowledge, but the representations are not knowledge itself, no more than a map is the territory it describes. The "knowledge acquisition bottleneck" is a misleading metaphor. It suggests that the problem of developing a knowledge base is to squeeze a large amount of already-formed concepts and relations through a narrow communication channel. In contrast, knowledge acquisition usually involves inventing new languages for modeling previously unarticulated experience.

Choosing and evaluating knowledge acquisition methods can be facilitated by shifting our perspective about the nature of knowledge engineering:

1) The primary concern of knowledge engineering is modeling systems in the world, not replicating how people think (a matter for psychology).

2) Knowledge-level descriptions (e.g., "this physician follows this diagnostic strategy") characterize human behavior in some social environment—what people say and do in particular situations, not stored, physical structures inside the head.

3) Modeling intelligent behavior is fraught with frame-of-reference confusions. We must tease apart the roles and points of view of human experts, mechanical devices they interact with, the social and physical environment, and observer-theoreticians (with their own interacting suite of recording devices, representations, and purposes).

The challenge to knowledge acquisition today is to clarify what we are doing (computer modeling), clarify the difficult problems (the nature of knowledge and representations), and reformulate our research program accordingly (how to collaborate with social scientists and users). I sketch out these ideas in this position statement.

Qualitative Process Modeling

In the past decade, we have studied knowledge bases and abstracted their designs, so we can describe what we are doing and devise methods to do it more clearly, reliably, and efficiently. Second generation expert systems separate out and make explicit the two processes that are modeled in every expert system (Clancey, 1983):

1) *a model of some system in the world* (the domain, e.g., a model of an electronic circuit);

2) *a model of reasoning processes* (the inference procedure, e.g., a diagnostic procedure).

These aspects of expert systems are reflected in two dominant, interacting areas of research, called *qualitative reasoning* and *generic expert systems*. The focus of qualitative reasoning (Bobrow, 1984) is to develop notations and calculi for modeling processes in the world. The focus of generic expert systems is develop task-specific representations and inference procedures (e.g., specific to diagnosis, configuration, scheduling, auditing, control) (Clancey, 1985). These complementary areas of research are integrated in expert systems and associated tools with enhanced capability for knowledge acquisition and explanation. Second generation expert system techniques produce a growing library of abstractions, enabling new programs to be constructed by reusing and refining existing representations and inference procedures (Marcus, 1988).

The content analysis involved in constructing generic expert systems is called "knowledge-level analysis." It contrasts with earlier emphasis on implementation-level distinctions (e.g., using rules vs. frames). Developing alternative representational notations (e.g., more formal conceptual structures (Sowa, 1984)) plays a secondary role. Questions about notations do not go away, but are recast in terms of tasks, domains, process representations, and model construction. Useful dimensions for describing expert systems include:

—the nature of the task (e.g., diagnosis vs. design);

—the nature of the system being modeled (e.g., an isolated, designed device vs. an open, biological system),

—how processes are represented (e.g., classification vs. simulation),

—the inference method for constructing a situation-specific model (e.g., contrasting alternatives on a blackboard vs. depth-first, incremental refinement of a single hypothesis),

—the macro-structure of the relational network used for describing the domain and inferential processes (e.g., hierarchies, state-transition networks, and compositions of these) (Clancey, 1989).

Questions of computer encoding are thus reformulated in terms of *process modeling methods* that separate process descriptions of the domain, inference, and communication (Clancey, in press).

In short, all knowledge bases contain models of systems in the world. A human expert serves as an informant about how a given system tends to behave, how it can be designed or controlled to generate desirable behaviors, and/or how it can be assembled or repaired. It follows that an expert system's performance can be evaluated in terms of the suitability of the model it constructs for the purpose at hand. For example, for medical diagnosis we need to look beyond the names of the diseases output by the program to determine whether the preferred diagnosis covers the symptoms that require explanation (Clancey, 1986). Previously, such consideration of completeness and consistency was reserved for programs using simulation or so-called "model-based reasoning." But all expert systems construct models and can be evaluated on this basis.

In summary, *qualitative reasoning* embraces modeling based on classifications (e.g., a taxonomy of disease processes), as well as modeling based on simulations (e.g., a behavioral simulation in the form of a causal network relating abnormal substances and processes internal to the system being modeled). From this second generation viewpoint, we can define knowledge engineering as a *methodology for modeling processes qualitatively*, in the form of relational networks describing causal, temporal, and spatial relations. Having shifted from the view that the knowledge base is a model of expert knowledge exclusively, we have no qualms about integrating qualitative and numeric models. We are belatedly discovering that many expert systems have done this all along. For example, SOPHIE used qualitative modeling to control and interpret a FORTRAN simulation of its electronic circuit (Brown, et al., 1982). SACON used simplified numeric equations to estimate stress and deflection, which were then abstracted to select programs that provide more detailed analysis (Bennett, et al., 1978)).

Why classification models are necessary

The knowledge engineering community's disparagement of classification in the 1980s went beyond the suggestion that it is not modeling. Many papers in the literature suggest that classification models are inferior to simulation models and can be entirely reduced to or compiled from them (e.g, De Kleer and Brown, 1984). According to this point of view, physicians talk in terms of syndromes and disease classifications because they do not understand the causal mechanisms causing these processes. A "real" model would reduce disease descriptions to descriptions of physical structure and function. For the most part these assumptions are false and belie a fundamental misunderstanding about the nature of system modeling and, more generally, how systems interact.

Disease descriptions characterize *the result of recurrent interaction* between an individual person and his or her environment. Consider for example tennis elbow. This syndrome cannot be causally explained in terms of processes lying exclusively within the person or within the environment. Rather it is a result of a pattern of interaction between the person and environment over time. As for any emergent effect, it can't be predicted, explained, or controlled by treating the person in isolation, or even by studying the person-environment system over short periods. It is a developmental effect, an adaptation in the person that reflects the history of his or her behavior in the world. The same claim can be made about the entire taxonomy of medical diseases—trauma, toxicity, infection, neoplasms, and congenital disorders—they are all descriptions of bodily processes after a history of recurrent interactions. Similar examples can be drawn from computer system failures; faults cannot be reduced to changes in a blueprint, but are in fact constantly introduced and prone to change in an open environment. A favorite story at Stanford's SUMEX-AIM is how system crashes were caused every fall when the first October rains wet the phone lines going to Santa Cruz, swamping the computer with spurious control-C's attempting to get its attention. Such problems aren't fixed by swapping boards.

The consequences of this systems-modeling perspective are more staggering than we might first imagine. Simply put, blueprints and functional diagrams of a device being modeled (including the human body), fail to capture emergent, historical effects of the system's interaction with its environment over time. If the device is *adaptively developing new structures* during its interactions with its environment, then a classification model is necessary in order to characterize how the device will behave and be internally organized over time. Such descriptions are necessary in order to describe the state of the device, to explain—historically—how it got into this configuration, and thus to provide a basis for modifying or controlling the system in some desired way (e.g., to prevent the tennis elbow from recurring). Biological systems are replete with examples of emergent structure; common examples are tree rings, the spirals of the nautilus shell, and the distribution of species over the landscape (Bateson, 1988).

In effect, a category jump has been made: The system we are now describing is the environment and the embedded device interacting over time, not the device in isolation. Thus, classification models constitute a level of system description, but they cannot be reduced to or mapped onto pre-existing physical structures in individual devices. As we move from blueprint-like structure-function models, we move from the domain of an isolated system to social, interactive, emergent processes. As Ryle warned us, we make a category mistake if we try to find the university in the members of colleges, the division in the parade of soldier battalions, or team spirit in specific "cricketing operations" (Ryle, 1949, p. 16). It is no coincidence that Ryle's examples all contrast social organizations with individuals or entities viewed in isolation. To suppose that classification models of how adapted behavior of a system-in-its-environment appears to an observer can be reduced to internal mechanisms of individual agents that existed before the interaction began is to make a category mistake.

We have to be careful in modeling complex, interactive systems like a computer, the human body, or a team of workers. We are interested not only in how a system works (its components and their purposes), but how its behavior develops in different interactional environments. This is precisely the province of the human expert, who can tell us what he has observed from experience, as he has participated in the system's operation. For different purposes, we may find it necessary to get the viewpoint of different observers, providing descriptions relative to different points of view (for further discussion, see (Clancey, 1991d; in preparation b)).

Knowledge and Representations

An observer ascribes knowledge to a human agent in order to describe and explain recurrent patterns of behavior in some environment. Knowledge-level descriptions (e.g., natural language grammars and problem-solving strategies) cannot be reduced to mechanisms in the body of individual agents; they are relative to the observer's point of view and characterize the total system of agent plus environment. In effect, there are several related claims:

—a knowledge representation such as a situation-action rule describes how someone behaves in some environment;

—as pattern descriptions, rules and scripts describe habits and routines that develop over time. Behavior patterns cannot be attributed to internal, stored structures that are first learned as theoretical descriptions (e.g., memorized facts and procedures) and subsequently control behavior in the manner of a template (by retrieval and mere application, in the manner of a tool that remains the same from situation to situation (Lave and Wenger, 1991)).

Part of the confusion in relating knowledge bases to human behavior is that we work backwards from our models to attribute properties of the computer to people. Observing the static nature of rules stored in a computer memory, we start explaining human behavior in terms of retrieving, matching and interpreting stored rules. We view human behavior as caused by symbolic structures. This is certainly true of computer system behavior, but it is a great leap to assume that it is literally true of people. Our representations have a great effect on how we see people, to the point we forget that an expert system is *just a model*, and that psychological claims prevalent in the early knowledge acquisition literature (Hayes-Roth, et al., 1983) are disputable.

Philosophical and psychological studies of memory, representations, and perception (see Clancey, 1991a; 1991b; 1991c) suggest radical shifts from the early knowledge-engineering points of view that knowledge acquisition is "transfer of expertise" (Davis and Lenat, 1982). Crucially, we must distinguish between representations out in the world (such as this book chapter and rules in an expert system), perceptual experiences (such as silently talking or singing to yourself, or visualizing something), and neural structures which are coming into being during our behavior.

We must not confuse representations of knowledge with whatever neural structures are in the brain coordinating our activity. A knowledge-level description, as a physical representation, must be expressed in some perceived medium. When we speak we are not translating internal representations of what our words mean, but creating the representations in our activity. Interpretable representations only exist physically in an observer's statements, drawings, computer programs, silent speech, etc.

Representing meaning is a subsequent perceptual act. In interpreting an already existing representation—that is, in using it—we perceive some structures and comment on what they mean. Representations, including knowledge representations, are always open to interpretation; their meaning is never fixed or defined, but always relative to an observer's frame of reference in the next act of interpretation (Agre, 1988). Thus, a second level of perceptual construction is interposed by the observer of the observer's representations (Clancey, 1991d).

Elaborating some implications, we find ourselves almost overwhelmed with reasons for doubting that a knowledge base can be associated with structures that were previously encoded in the head of the expert:

❑ Knowledge-level descriptions are attributions made by an observer (the knowledge engineer), involving selective interactions with the agent (the expert), personal perceptions, and point of view;

❑ Knowledge-level descriptions abstract a sequence of behaviors (what the expert does and says in the course of solving a sequence of related problem examples), not single, moment-by-moment responses;

❑ Descriptions of the device being modeled and inferential processes are informed by the expert's observations and problem-solving behavior, but they are not primarily intended to be the expert's "mental models" or psychological explanations of behavior;

❑ To the extent that the processes people follow in gathering data to solve a problem and taking action in the world are intended to be simulated by the expert system, these descriptions always model the combined, usually social, system (how the expert interacts with his or her environment);

❑ Meaning is not fixed or stored in the brain. Knowledge-level descriptions have an open interpretation, dependent on the point of view of the observer of the representation, in the course of his or her ongoing activities;

❑ People express knowledge-level descriptions in perceptual space (e.g., on paper or in a computer file), so they can be subsequently reperceived and interpreted (Clancey, in preparation). Human use of representations (e.g., reading this article) involves new conceptualization, not syntactic manipulation of definitions and meaning templates, which is all that today’s computer programs can do (Clancey, 1991a);

❑ Designing knowledge representations is the province of the knowledge engineer and AI researcher. The human expert, despite often being a theoretician of his or her own behavior, in general does not bring representational languages ready-made to the knowledge acquisition session. Otherwise, there would be no need for knowledge representation research.

In light of this perspective, it is illuminating to reinterpret Newell's comments about the knowledge-level (Newell, 1982) *(reinterpretations in italics):*

> The knowledge level is not realized as a state-like physical structure, "running counter to the common feature at all levels of a passive medium." (p. 105) "It seems preferable to avoid calling the body of knowledge a memory." (p. 101) "The total system (i.e., the dyad of the observing and the observed agents) runs without there being any physical structure that is the knowledge." (p. 107)
>
> *—Knowledge isn’t a substance that behind the scenes causally drives human behavior. A knowledge-level pattern, such as a natural language grammar characterizes the product of interaction, how behavior routines and the world appear; it doesn’t describe structures in the head. Put another way, the neural structures coordinating perception and action in people come into being in the course of interaction itself. Thus, new knowledge (capability to speak, perceive, and act) develops as new coordinations in the course of behavior itself; people are not automatons rotely executing stored programs.*
>
> Knowledge can only be "imagined as the result of interpretive processes operating on symbolic expressions." (p. 105)
>
> *—When we comment on (represent) the meaning of perceived structures, in effect claiming that they are symbolic, we reorient our behavior (e.g., reading a map, following instructions). Interpretation of representations by people is always perceptual, involving new conceptualization, not syntactic interpretation as in an inference engine. This gives us new capability to act, which we call knowledge.*
>
> "Knowledge of the world cannot be captured in a finite structure." (p. 107) "Knowledge can only be created dynamically in time." (p. 108)
>
> *—Every human behavior is an adaptation; every perception, thought, and action is a generalization (Clancey, 1991b). Words, meanings, and understandings are not merely retrieved or syntactically combined. Coordinations and capabilities are always at some level new; they are constantly constructed out of previous coordinations. Again, we must distinguish between the product of knowing (a spoken sentence) and the process by which neural structures are selected and recomposed. People needn’t store away descriptions of behaviors; our capability to speak and act is created dynamically in time.*
>
> "One way of viewing the knowledge level is as the attempt to build as good a model of an agent's behavior as possible based on information external to the agent." (p. 109)
>
> *—The knowledge engineer's knowledge-level description of the expert emphasizes the expert's awareness and use of materials and circumstances in the environment; that is, it accounts for behavior in terms of interaction between agent and environment. Reasoning procedures and domain models describe how experts behave, as well as recurrent processes in their environment. Knowledge bases are models; no claims need be made about internal structures in the agent.*

As Newell says, knowledge can be represented, but it is “never actually in hand.” Each statement by the observer captures what he needs to say at any point in time, and each such statement is later interpretable in different ways. We must work against the common sense tendency to rationalize observed behavior in terms of physical representations of goals, meanings, intentions, and assumptions that supposingly exist inside the head of the agents before behavior begins. People can of course represent their goals and assumptions, and this of course influences their behavior. But all human behavior—including uttering such representations—is immediate, without requiring intermediate plans or other semantic schemas that model what we are about to say or do. When an observer describes an intelligent agent, a distinction needs to be drawn between *knowledge as a capacity* ascribed to the agent (dynamically changing through interaction with the environment) and *the observer's representations of this capacity* (perceivable structures, open for interpretation). Hence, we may be ready to return to and build upon Ryle's famous distinction between knowing how (a capacity to perform some action) and knowing that (a representation). The capacity to perform cannot be reduced to (mechanistically replaced by) knowledge-level descriptions of how the performance *appears*.

Perhaps the strongest claim is that a machine that syntactically manipulates representations can model human behavior, but as an agent, an expert system isn't capable of what the human brain allows in flexibility and creativity. This isn't something that can fixed by adding more representations, but requires inventing a new kind of mechanism that doesn't rely on stored models or programs (Clancey, in preparation a). This places a premium on understanding the differences between today's expert systems and human capability, and exploring uses for computers beyond automation of reasoning.

Implications for Socio-Technical System Design

What are the implications for expert system design and knowledge acquisition if human reasoning is not produced by interpreting stored representations? First, we must adopt a different way of talking about our programs. They are only models, not intelligent beings. We are not modeling structures in the expert's head, though we will certainly continue to pay close attention to how experts talk and what representations they use (e.g., diagrams, logbooks, notational shorthand, calculi). We are free to incorporate different kinds of models in whatever combination is useful for the task at hand, no longer bound to vaguely relate knowledge bases to expert methods. Adopting the systems-modeling perspective suggests that numeric approaches should be freely integrated (e.g., linear programming, Bayesian statistics).

But more radical changes to knowledge engineering are required. In developing expert systems, we must reconsider how human work relates to computer models. To restate some claims made above:

❑ **A representation is not equivalent to knowledge.**

A representation of what a person knows is just a model of his or her knowledge, a representation of a capacity. Knowledge cannot be reduced to (fully captured by) a body of representations. Knowledge cannot be inventoried.

❑ **The meaning of a representation cannot be made explicit.**

Meaning can be represented, but it cannot be defined once and for all, captured fully by representations. The meaning of a representation is open, though there are culturally stable representations of meaning (e.g., word senses).

❑ **The context in which a program is used cannot be made explicit.**

Context can be represented, but the world cannot be objectively and exhaustively described; cultural or social circumstances cannot be reduced to a set of facts and procedures (Lave, 1988).

One way of summarizing this is "practice cannot be reduced to theory." This contrasts with the familiar idea that theoretical descriptions are a kind of ideal, but the world is a messy place. In effect, by saying that human behavior isn't driven by stored theoretical descriptions (e.g., formal procedures, rules, or models), we are saying that models of behavior and the world always selectively abstract and give a limited impression of human capabilities. It is the unspecifiable "messiness" of the neural system—becoming organized in new ways at the time of interaction itself—which gives human behavior its robust, always adaptive character.

The limitations of scientific models based on pattern descriptions has also been brought to the forefront by the invention of chaos models (Gleick, 1987, p. 6):

> Twentieth century science will be remembered for just three things: relativity, quantum mechanics, and chaos.... Like the first two revolutions, chaos cuts away at the tenets of Newton's physics. As one physicist put it: "Relativity eliminated the Newtonian illusion of absolute space and time; quantum theory eliminated the Newtonian dream of a controllable measurement process; and chaos eliminates the Laplacian fantasy of deterministic predictability." Of the three, the revolution of chaos applies to the universe we see and touch, to objects at human scale. Everyday experience and real pictures in the world have become legitimate targets for inquiry.

Strikingly, at the level of workplace analysis, both knowledge engineering and ethnography have opened up everyday experience as a target of inquiry (Lave, 1988). But like the physicists, we must make some new distinctions between our models and the phenomena of study. We must distinguish between activities, patterns, and theories:

Social activities and physical phenomena: The world being modeled has an inviolable nature; it cannot be exhaustively described. We can model the world, but we can always go back to find new perspectives for describing what we are modeling, usually involving new perspectives on what constitutes information (data), new languages for modeling, and new perspectives on the purpose for constructing models.

Design and interaction patterns: Rules, classifications, scripts, grammars, structure-function models, causal state-transition networks, metaphors, statistics, etc. are useful for describing complex designs and social systems. Models are especially useful for creating new designs (Alexander, et al., 1977), diagnosing and repairing undesired situations, and teaching. But we must remember that models (notably formal specifications) remove us from the world we are attempting to understand and influence. In the design process, for example, we must develop disciplined means of relating tools to the context of use.

Social-psychological theories: At another level, we develop theories about why the models we create are valid, why these representations have been constructed and not others. For example, the idea that the purpose for using a model determines what kind of model is desirable is part of knowledge engineering theory. In general, metatheoretical considerations help us organize our modeling techniques into a coherent methodology. For example, having related modeling techniques to domains (Clancey, 1986), we might go back to the world of artifacts and social activities to flesh out our repertoire by attempting to model new domains. In general, to be effective, knowledge engineering requires more extensive, integrated theories of work, collaboration, communication, understanding, creativity, routines, perception, and representations.

One implication of these distinctions is that researchers should make clear whether they are providing practical knowledge acquisition tools or focusing instead on theories and new modeling techniques. Providing tools requires more careful attention to the social setting in which expert systems are used, focusing on how teams of people interact to solve problems and how job aids can facilitate this interaction.

Studying the nature of intelligence will continue to involve knowledge-level analyses, for this is the leverage that cognitive science provides over neurobiology. However, a clear separation should be made between knowledge-level descriptions and physical mechanisms. The idea that human-equivalent behavior could be generated by interpreting stored programs that predescribe the world and ways of behaving must be abandoned, for this view confounds descriptions an observer might make with physical mechanisms inside the agent.

Researchers can commit to both practical knowledge engineering and the study of intelligence, as surely both feed into each other. However, the practical needs of tool users and the difference between knowledge bases and the human mind require a more explicit commitment than before, otherwise evaluation and choice of methods will be confused.

To elaborate on what can be done today, I will discuss two recommendations for designing expert systems:

1) Don’t just deliver technology; collaborate with social scientists and users on site in tightly-incremental designs.

2) Facilitate, don’t just automate conversations.

Collaborate with users in tightly-incremental designs

Social-technical systems must evolve; they are deterministic, but not predictable, and therefore cannot be controlled. We can design organizations, but we cannot control how people will work together, how they will actually accomplish what they need to do. At another level, this means that we cannot control or strictly predict how people will construct goals, sources of information, or new tools. When we supply technology, we cannot predict all the nuances of how the tools might be exploited or how they might change the social interactions and roles (Zuboff, 1988; Greenbaum and Kyng, 1991; Wenger, 1991).

One implication is that knowledge engineering splits between the attempt to invent new theoretically interesting uses of computers and the attempt to deliver useful tools for industry, schools, professionals in the short term, while furthering our theoretical understanding. This “action-oriented research” can be viewed as basic research on the problem of how to design useful tools in partnership with users on the job. Researchers focusing on these problems believe that the fundamental problems are not just in the realm of technology, but in understanding what workers are doing and in changing work practice (Zuboff, 1988; Ehn, 1988; Wynn, 1991).

Research shifts to the *design process:* learning how to discuss designs with non-technical people, finding out how work really gets done, promoting invention, resolving organizational paralysis (Bannon, 1991). Central design questions include:

1) How should new technology be co-designed with users?

2) What program features would enhance apprenticeship, non-routine problem solving, and innovation?

3) What changes to organizational structures are required by and would enhance the new tools?

Recently, anthropologists, sociolinguists, and human factors specialists have been collaborating to invent new ways of working with users, new uses of computers, and new organizational structures (e.g., Zuboff, 1988; Kukla, et al., 1990; Greenbaum and Kyng, 1991; Hughes, et al., 1991). The role of ethnography is to provide a global view of the workplace, to keep tool design integrated into the dynamics of the workplace, and to know what other tools should be built and how they are related to worker identity and role. Social scientists in effect help to keep the project honest. They ask, “Are we solving the most pressing problem? How does our technology relate to users’ priorities? What non-technical factors could lead to failure?” This is similar to a “market analysis,” but based on looking at how people work together—more like investigative journalism than psychological experimentation or surveys.

A key idea is rapid, incremental development in the context of use. In effect, this entails redistributing responsibility for design. Such a shift is facilitated by good prototyping tools, so programmers are less committed to early designs and other people have control over design decisions (e.g., users, graphic designers, managers). The role of prototyping is not just a way of making programming efficient, but a means of keeping programmers and users open-minded, ameliorating the investment in tedious work to implement any given design. In effect, program design needs to be more like architectural sketching than laying bricks in concrete. We need the interface equivalent flexibility of moving around walls and furniture, not nailing and sawing wood. This is the promise of task-specific programming environments (Clancey and Barbanson, 1991).

A new role for knowledge engineering is to help ethnographers organize and model workplace observations. Ethnography could benefit from a process-modeling language (scripts, transition networks) for describing how people interact. Notably, such models transcend individual points of view. They describe what coordination between people accomplishes as a whole, not individual "reasoning." They include pattern descriptions that many people in the workplace itself might not recognize (Jordan and Alpert, 1991). They are patterns of interactions, not templates or formal procedures. In effect, we can use qualitative modeling techniques to analyze and share ethnographic data—to model workplace interactions—without making commitments to putting models in computer tools for workers.

Specifically, qualitative work process models could:

❏ Organize data by classifications, state-transition networks, and structure-function hierarchies (e.g., Jordan's "typified action sequences").

❏ Model job functions and schedules, representation manipulation (e.g., how logbooks are modified and shared), and interaction patterns. That is, qualitative modeling can be used in the workplace to model physical systems, reasoning, and communication processes (Clancey, in press).

❏ Represent agent roles and interaction strategies. In knowledge representation systems, we have developed languages for describing roles that people play and strategies they follow.

Such formal models could complement more prosaic ethnographic descriptions, for example, by providing multiple indices to a video library illustrating workplace practice. In effect, representational languages and calculi developed for knowledge engineering can be used broadly to model the interaction of social, physical, and technological systems.

Facilitate, don't just automate conversations

Beyond new approaches to design, we should consider radical changes in how computers are used. Most computer programmers emphasize automation. They build tools exclusively around formal descriptions of work ("functionalism" (Kukla, et al., 1990)). Computer tools have an individual orientation (the workstation view). These biases are reinforced by the laboratory design approach, in which basic research occurs apart from the application setting; controls and idealizations distort the nature of practice. Many of these ideas were first articulated in the AI community by Winograd and Flores (1986), who added the subtitle to their book, "A new foundation for design" to emphasize the relation between a changed conception of human reasoning and new ideas of how computers can be used.

The information-processing view of people is quite idealized. People are usually described one-dimensionally—assumed to be on-task, rational, dedicated, and loyal to the company. Although knowledge engineers pay lip service to such ideas as "breaking down barriers to communication," they focus exclusively on access to information, leaving out issues of identity and membership in the organization (Wenger, 1990). What interactions occur outside the web of information-processing computers and telecommunications links? Work schedule, salaries and job scales, war stories, role-defined "knowledge-making rights" (Eckert, 1989) are all important workplace considerations that computer tools might take into account.

As an example, consider Kukla et al.'s (1990) study and designs for process control communication in a Monsanto plant. Kukla's view of work is dynamic, always non-routine, and intricately formed by a web of interactions greatly distributed in space. Following Winograd and Flores' advice, Kukla modeled conversational interactions in great detail. In contrast with traditional knowledge engineering, Kukla's proposed communication tool designs take into account that people dynamically define what their tasks are and reconceive what constitutes information for doing their job.

But Kukla's view is always oriented towards problem-solving at the manufacturing task level. People are only described as they exist "on task," without any sense of the dynamics of how roles get defined, how new people are brought on board, how conflicting interpretations are resolved, etc. Kukla designs are claimed to promote innovation, but he doesn't say how, except to say that the right people are put in touch with each other, and they can show each other what is happening (different views of the work), at critical times. How does learning occur? How are contradictory goals of different organizations reconciled? (Kling, 1991) Kukla's proposed tools for the Monsanto workers are strikingly different from most "automate everything" systems. But by providing more details and theoretical descriptions of what is happening, we might further justify and improve these designs. A learning perspective would focus more on how new practices are introduced, rather than just how serious events are handled.

For example, we should analyze what changes in people's interactions as a result of working through a difficult situation together. In effect, we are *designing for communities of practice, not information processors* (Wynn, 1991; Wenger, 1990).

Given the distinctions between human knowledge, practice, and representations I have laid out, we might reformulate how we view qualitative modeling. Example shifts in perspective:

❏ **Represent patterns causing breakdown of models.**

That is, to facilitate group interpretation of past work, represent the cases on which the model fails and rationalizations for the failures. View failure annotations as ways of representing the boundaries of a model; recognize that such boundaries always exist and that modeling them is important for users.

❏ **Don't attempt to exhaustively model the world in terms of patterns.**

Rather than attempting to build an omniscient program, capitalize on the program's inability to model every situation. For example, detect when modeling exceptions occur and sound an alarm (Byrnes, et al., 1990).

❏ **Provide modeling tools for students and workers to reflect on their practices.**

View the computer tool not as doing someone's job for them, but as a means for them to represent typical work practices, as well as the details of specific situations, and reflect upon them (Rodolitz and Clancey, 1989).

❏ **Use computer models as tools for mediating conversations.**

Rather than constructing a tutor to talk to an individual student, conceive of the simulation model as a means for students to experiment and explain things to each other (Roschelle, 1990). How can an expert system be used to facilitate a conversation between a sales person and a client? Between the sales person and the in-house product designers?

To use expert systems appropriately, we must respect how representations are continuously reinterpreted and created in social interactions. We must abandon the idea that the computer model is a kind of "correct," once-and-for-all view of the world. The representations people put in a knowledge base are as much for people as for the program. We must take into account how people continuously construct and reinterpret their own models in the course of their work (Wynn, 1991). Ethnographic studies (Linde, 1991; Jordan and Alpert, 1991; Kukla, et al, 1990) suggest that computer tools might be based on the following considerations:

❏ **Enable forms of sharing:**

—make resources (human and artifacts) more available;

—allow people to see, copy, incorporate, possess, modify, and become responsible for work in new ways.

❏ **Make tools accessible to everyone in the group:**

—support overlapping responsibilities;

—relate contributions, don't isolate jobs;

—accommodate novices and experts, people familiar or not with everyday situations.

❏ **Map designs to problem-solving and innovation phases:**

—phases include orient, explore, collaborate, coordinate, take action;

—allow non-routine processes to remain ad-hoc; allow the invention and flexibility required for dealing with difficult, emergency situations to still take place.

Conclusion

Several ideas interweave in this analysis: Today's computer models are limited in capability relative to people; qualitative modeling provides a new basis for tools of value to business, science, and engineering; and social science perspectives change the interpersonal dynamics of software design. In essence, a new view of people relative to computer models yields a new view of what tools can be, and hence a new view of the tool design process (Winograd and Flores, 1986). In this respect, the rhetoric of Scandinavian design approaches (e.g, Ehn, 1988) appears less harsh than it might first appear. Knowledge engineers are called to participate with social scientists and workers in the co-design of the workplace and tools for enhancing worker capabilities. The emphasis is on augmenting human capabilities, not merely automating what they do. Significantly, this must be done in the context of use, to maintain connection to non-technical factors such as ownership of ideas, based on the worker's sense of identity and membership in a community.

In many respects, this research has just begun. Some of the open issues include:

❏ **An evolutionary view of organizational learning:** What is the interplay of individual insights, group trends, boundaries between communities, multiple membership, and forms and rights of participation (Bartlett, 1932; Wenger, 1990; Eckert, 1989)?

❏ **Computer tools to facilitate learning on the job:** What designs provide for shared workspaces, allow improvisation, and coordinate contributions? How can “intelligent tutoring” systems be incorporated in job performance aids?

- ❏ **Representation and meaning construction:** What are heuristics for promoting creativity, encouraging creation and distribution of design stories? What are new artifacts for communication (e.g., electronic blackboards)?

In effect, knowledge engineering moves radically from its original concern in “acquiring and representing expert knowledge” to the larger arena of social and interactional issues involved in collaboration and invention in everyday work. We shift from the idea that a glass box design is an inherent property of a device, to realize that transparency is relative to the observer’s point of view, and this depends on cultural setting (Wenger, 1990). We shift from the idea that computer models are equivalent to habits and skills; rather *as representations* they play a key role in reflection and hence learning new ways of seeing and behaving (Schön, 1987). We shift from the idea that goals, meaning, and information are fixed entities that are inherent in a task, to helping people in their constant, everyday efforts to construct their mutual roles, contributions, and identity (Wynn, 1991). In all this, we see the role of knowledge engineering not as “capturing knowledge” in a program that is delivered by technicians to users. Rather, we seek to develop tools that help people in a community, in their everyday practice of *creating* new understandings and capabilities, new forms of knowledge.

Acknowledgments

A revised version of this paper will appear in *The International Journal of Intelligent Systems*, special issue on knowledge acquisition, edited by Ken Ford, and a book published by J.Wiley; This paper originally appeared without the Section 4 as ”The knowledge level reinterpreted: Modeling how systems interact,” *Machine Learning* **4**, 287-293, December 1989. John McDermott and Brigitte Jordan provided useful suggestions for improving this version. Funding has been provided in part by gifts from the Digitial Equipment Corporation and the Xerox Foundation.

References

Agre, P. 1988. *Writing and representation.* Unpublished MIT Technical Report.

Alexander, C., et al. 1977. *A Pattern Language.* New York: Oxford University Press.

Bannon, L. 1991. From human factors to human actors. In J. Greenbaum and M. Kyng (eds), *Design at Work: Cooperative design of computer systems*. Hillsdale, NJ: Lawrence Erlbaum Associates, pps. 25-44.

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

Bateson, G.1988. *Mind and Nature: A necessary unity.* New York: Bantam.

Bennett, J. Creary, L. Engelmore, R., and Melosh, R. 1978. *SACON: A knowledge-based consultant for structural analysis.* STAN-CS-78-699 and HPP Memo 78-23, Stanford University, CA, September.

Bobrow, D. G. 1984. Qualitative reasoning about physical systems: An introduction.*Artificial Intelligence,* **24**(1-4):1-6.

Brown, J.S., Burton, R.R., and De Kleer, J. 1982. Pedagogical, natural language, and knowledge engineering techniques in SOPHIE I, II, and III. In: D. Sleeman and J.S. Brown (eds), *Intelligent Tutoring Systems* (Academic Press, London), pp. 227-282.

Byrnes, E. Campfield, T., Henry, N. and Waldman, S. 1990. Inspector: An expert system for monitoring world-wide trading activities in foreign exchange. *AI Review,* **3** (July/August):9-16.

Clancey, W.J. 1983. The advantages of abstract control knowledge in expert system design. *Proceedings of the National Conference on Artificial Intelligence,* pp. 74-78.

Clancey, W.J. 1985. Heuristic classification. *Artificial Intelligence,* **27**:289-350.

Clancey, W. J. 1986. Qualitative student models. In J. F. Traub (ed), *Annual Review of Computer Science*. Palo Alto: Annual Review Inc., pp. 381-450.

Clancey, W. J. 1989. Viewing knowledge bases as qualitative models. *IEEE Expert,* (Summer 1989):9-23.

Clancey, W.J. 1991a. Why today’s computers don’t learn the way people do. In P.A. Flach and R.A. Meersman (eds), *Future Directions in Artificial Intelligence.* Amsterdam: Elsevier, pp. 53-62.

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

Clancey, W.J. 1991c. Situated cognition: Stepping out of representational flatland. *AI Communications,* **4**(2/3):107-112.

Clancey, W.J. 1991d. The frame of reference problem in the design of intelligent machines. In K. vanLehn (ed), *Architectures for Intelligence: The Twenty-Second Carnegie Symposium on Cognition*, Hillsdale: Lawrence Erlbaum Associates, pp. 357-424.

Clancey, W.J. in press. Model construction operators. To appear in *Artificial Intelligence.*

Clancey, W.J. in preparation a. Interactive control structures: Evidence for a compositional neural architecture.

Clancey, W.J. in preparation b. A Boy Scout, Toto, and a bird: How situated cognition is different from situated robotics. To appear in a special issue of the *AI Magazine*.

Clancey, W.J. and Barbanson, M. 1991. Using the system-model-operator metaphor for knowledge acquisition. *IEEE Expert*, **6**(5): 61-65.

Davis R. and Lenat, D.B. 1982. Knowledge-Based Systems in Artificial Intelligence. New York: McGraw Hill.

De Kleer, J. and Brown, J.S. (1984) A qualitative physics based on confluences.*Artificial Intelligence,* **24**(1-4), 7-84.

Eckert, P. 1989. *Jocks and Burnouts: Social Categories and Identity in High School.* New York: Teachers College, Columbia University.

Ehn, P. 1988. *Work-Oriented Design of Computer Artifacts.* Stockholm: Arbeslivscentrum.

Floyd, C. 1987. Outline of a paradigm shift in software engineering. In Bjerknes, et al., (eds) *Computers and Democracy—A Scandinavian Challenge,* p. 197.

Gleick, J. 1987. *Chaos: Making a New Science.* New York: Viking.

Greenbaum J. and Kyng, M. 1991. *Design at Work: Cooperative design of computer systems*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Hayes-Roth, F., Waterman, D., and Lenat, D. (eds) 1983. *Building Expert Systems*. New York: Addison-Wesley.

Hughes, J. Randall, D., and Shapiro, D. 1991. CSCW: Discipline or Paradigm? A sociological perspective. In L. Bannon, M. Robinson, and K. Schmidt (eds), *Proceedings of the Second European Conference on Computer-Supported Coooperative Work.* Amsterdam, pp. 309-323.

Jordan, J. and Alpert, B. 1991. *Technology and Social Interaction*, Xerox-PARC Technical Report.

Kling, R. 1991. Cooperation, coordination, and control in computer-suppored work. *Communications of the ACM*, **34**(12):83-88.

Kukla, C.D., Clemens, E.A., Morse, R.S., and Cash, D. 1990. An approach to designing effective manufacturing systems. To appear in *Technology and the Future of Work.*

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

Lave, J. and Wenger, E. 1991. *Situated Learning: Legitimate Peripheral Participation.* Cambridge: Cambridge University Press.

Linde, C. 1991. What’s next? The social and technological management of meetings. *Pragmatics,* **1**:297-318.

Marcus, S. 1988. *Automating Knowledge Acquisition for Expert Systems.* Boston: Kluwer.

Newell, A. 1982. The knowledge level. *Artificial Intelligence.* **18**(1):87-127.

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

Roschelle, J. 1990. *Designing for conversations.* Presented at the AERA Symposium on Dynamic Diagrams for Model-Based Science Learning, San Francisco, April.

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

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

Sowa, J. 1984. *Conceptual structures*. Reading: Addison-Wesley.

Wenger, E. 1990. *Toward a theory of cultural transparency: Elements of a social discourse of the visible and the invisible.* PhD Dissertation in Information and Computer Science, University of California, Irvine.

Winograd, T. and Flores, F. 1986. *Understanding Computers and Cognition: A New Foundation for Design.* Norwood: Ablex.

Wynn, E. 1991. Taking Practice Seriously. In J. Greenbaum and M. Kyng (eds), *Design at Work: Cooperative design of computer systems*. Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 45-64.

Zuboff, S. 1988. *In the Age of the Smart Machine: The future of work and power.* New York: Basic Books, Inc.