

Developing Learning Technology in Practice

William J. Clancey

Institute for Research on Learning
2550 Hanover Street, Palo Alto CA 94304
Bill_Clancey@IRL.com

Draft: October 19, 1994; Revised June 6, 1995

Clancey, W. J. 1998. Developing learning technology in practice. In C. Bloom and R. Bowen Loftin (Eds.), Facilitating the Development and Use of Interactive Learning Environments (pp. 3–20). Hillsdale, NJ: Lawrence Erlbaum Associates.

Introduction

The US West Workshop on Intelligent Tutoring Systems (ITS)¹ examined the state of the art of instructional programs, focusing on the design and deployment of systems using Artificial Intelligence programming methods. Our objective was to appraise the progress in bringing advanced research ideas to practice and thus to understand the barriers and opportunities for using ITS technology in industry today.

ITSs were presented from US West, Nynex, Bellcore, NASA, Galaxy Scientific, Honeywell, and the universities of Massachusetts, Colorado, and Pittsburgh. Applications included troubleshooting electronic circuits, customer telecommunications operations, COBOL programming, space shuttle payload operation, introductory kinematics, cardiac arrest diagnosis and treatment, and kitchen design. These programs are distinguished from conventional computer-based training (CBT) by the use of qualitative modeling techniques to represent subject material, problem-solving procedures, interactive teaching procedures, and/or a model of the student's knowledge (Clancey, 1986, 1987 ; Self, 1988).

This chapter is based on a summary talk, prepared during the meeting for presentation the afternoon of the last day to stimulate group discussion. The stance is critical and yet confident about a new beginning: As ITS research struggles in a world of more limited research funding, affordable multimedia technology makes it possible to realize the early 1970s vision of intelligent teaching assistants. My primary observation is that ITS technologists have a dual objective: To develop flashy multimedia models that can be used for instruction and to fundamentally change the practice of instructional design. This second objective—promoting organizational change—is rarely mentioned and lies outside the expertise of most academic design teams. But reflecting the maturity of the technical methods, presentations at the workshop showed an emerging interest in effective design processes for everyday business, school, and government settings. Researchers are beginning to consider how difficulties in getting people to understand

¹The meeting was sponsored by US West and held in Boulder, Colorado, July 26-28, 1993. The original title of the workshop referred to ITS, so this term has been used in this chapter. Increasingly developers have preferred to shift the focus to learning and the student, emphasize interaction, and include non-technical tools, facilities, etc. Hence, the term “interactive learning environments” is more often used today. The focus of this paper is specifically programs developed by AI researchers, which universally were called “intelligent tutoring systems” during the decade beginning around 1977.

and adopt ITS methods are problems of organizational change, and not just technical limitations.

On the other hand, researchers understand that the greatest value of ITS technology will not be realized if AI methods are used merely to replace existing lectures or computer-based training, and consequently evaluated solely by the standards of existing training. Indeed, many ITS designers in corporations are pressed to adopt the metrics of cost and efficiency that fit a transfer view of learning and static view of organizational knowledge. If the traditional views of learning and assessing instructional methods are applied alone, the value of ITS approaches for changing classroom instruction may not be accepted or realized.

Creatively exploiting ITS technology—to change the practice of instructional design—requires a better understanding of how models relate to human knowledge. Relating the insights of the cognitive, computational, and social sciences involves changing how scientists, corporate trainers, and managers alike think about models, work, and computer tools. Broadly speaking, models comprise simulations, subject matter taxonomies, scientific laws, equipment operational procedures, and corporate regulations and policies. Instructional designers and developers of performance support tools must better understand how the interpretation in practice of such *descriptions* is pervaded by social concerns and values (Ehn, 1988; Floyd, 1987; Greenbaum and Kyng, 1991). Social conceptions of identity and assessment influence choices people make about what tools to use, methods for gaining information, and who should be involved in projects. Judgments about ideas reflect social allegiances, and not just the technical needs of work. This broader perspective on how participation and practice relate to technology moves ITS research well beyond the original focus in the 1970s on how to create models that represent different kinds of processes in the world. Inquiring about what models should be created and who should create and use them, we must consider new research partnerships, new design processes, and new computational methods for facilitating, rather than only automating conversations.

How do we move technology into the mainstream?

Participants in the US West workshop experienced a striking paradox: Their instructional programs are based on methods developed over nearly 25 years in internationally-known computer science and psychology research laboratories, but effectively no one in the multi-billion dollar industry of “corporate training” uses this

methodology. Instead, “computer-based training” is barely beyond the page-turning quiz generators of the 1960s, giving all computer approaches a problematic reputation.

Technically, there is a substantial gap between academic laboratory software and most training systems used in business today. Even off-the-shelf multimedia tools are at least a decade behind ITS representation and modeling techniques. Fortunately, the movement to object-oriented or “component-based” commercial software provides a means for sharing tools and models. But both the technological and collaboration infrastructure are still misaligned in these two cultures: Industry is only now accepting the windows and menus interface familiar to scientists in 1980, and still views Lisp, an established tool for three decades in academia, as a foreign language. Research funding was often conceived by corporations as throwing water on someone else’s garden, without establishing ways of learning in depth new methods and perspectives (epitomized by Xerox’s failure to commercialize the personal computer). Ironically, funding for AI in general contracted in the 1980s, under a general complaint that the work was “over-hyped” and not relevant to pressing problems. Such complaints bring out the real mismatch between past research and everyday business, and the gap in current understanding:

- Does industry understand the generality of qualitative modeling methodology to science and engineering, or is the ITS approach viewed as just a smarter page turner?
- When development costs for ITS are appraised as being too high, are the multiple uses and reusability of models considered?
- Is it surprising that ITS programs are not immediately embraced by users, when participation in the projects has not included conventional instructional designers, graphic artists, workers, and managers?

There are many reasons why the ITS methods of the mid-70s are not in use today. Indeed, the reasons are overwhelming:

- The computational methods are new, a radical departure from numeric programming.
- Graphic presentations required a change in hardware and software from traditional suppliers (especially IBM).
- The use of workstations in research applications predated their availability in industry by nearly 15 years (when prices dropped by more than ten-fold).

- The view of knowledge and learning in 1970s cognitive science (and embedded in the design of ITS) is not congruent with the views of workers and managers (Nonaka, 1991).
- A “delivery” mentality for software engineering in academia and industry alike prevents a participatory relation between researchers and their sponsors.
- In the late 1980s, the workforce became more distributed, with separate business units and “integrated” (non-specialized) employees, making centralized classroom training less appropriate.

Of all these considerations, one of the most important is the shift in how knowledge and learning are conceived. By the now well-known, “situated learning” perspective, learning is viewed as something occurring all the time and having a tacit component (Lave and Wenger, 1991; Clancey, in press, in preparation). Concepts are not merely words and definitions, but ways of coordinating what we see and how we move. Understanding is spatially, perceptually, and socially embedded in activity. Activities are not merely tasks, but roles, identities, and ways of choreographing interpersonal interaction. Problems are constructed by participants, not merely given. “Trouble” is defined in conversations about values, how assessments will be made, and who is participating. In these terms, documents and tools are not specified and given, but open to interpretation, having new meanings and uses in changing circumstances—according to workers’ experience, not merely packaged by teachers and rotely digested. Communication with co-workers is viewed as central, especially by informal relationships of friendship, developing through meetings and chance encounters.

None of this makes the articulation of principles, rules, and policies in written text irrelevant. Rather, this “situated cognition” analysis reveals how such descriptions of the world and behavior are created, in what sense they are shared, and how they are given meaning in practice. The result is that both creation of work representations or tools and their use must be understood in the context of work activity. Put concisely, one participant at the US West workshop said, “Classroom learning should be modeled after workplace learning, not vice versa.” Crucially, we don’t want to fall into an either-or mentality and impose one view on another, such as bringing CBT to the desktop or bringing “on the job training” to the classroom (indeed, this occurs!). Instead, we must ask, how do formal descriptions and training facilitate everyday recoordination and reinterpretation? (Wenger, 1990) The workplace is not just a context for learning; we are not shifting an activity from one place to another. Rather, we must reconceive what is being learned—beyond the curriculum—what problem solving is actually done on the

job? Is a worker’s problem to learn a rule, to interpret it, or to improvise around it? Especially, work must be viewed systemically: How does one person’s solution create a problem for someone else? Again, the shift is from formalized procedures applied in narrow functional contexts to conversation, anticipation, broad understanding, and negotiation.

Researchers with systems in use are aware that the issues aren’t all technological. Broadly speaking, instructional design must include understanding and configuring interactions that occur in practice between people, systems (and tools in general), and facilities. Table 1 summarizes the shift from a technology-centered perspective to one viewing the total system of interactions in practice.

Tool Design View (Technology -Centered)	Design for Everyday Use (Practice-Centered)
Teachers and students as subjects	Users as partners in multi-disciplinary design teams (participatory design)
Delivering a program in a computer box	Total system perspective, designing the context of use: organization, facilities, and information-processing
Promoting research interests	Providing cost-effective solutions for real problems
Automating human roles (teacher in a box) (represent what’s routine)	Facilitating conversations between people (assistance for detecting & resolving trouble)
Knowledge as formalized subject matter (models & vocabulary); learning as transfer to an individual.	Promoting everyday learning: conflict resolution and interpretation of policies (social construction of knowledge).
Transparency as an objective property of data structures or graphic designs.	Transparency and ease of use as a relation between an artifact or representation and a community of practice.

Table 1: Shift from ITS technology-centered development to design for everyday use.

Again, we need a “both-and” mentality: The perspective on the left side is not irrelevant; it is useful and important. For example, technologists surely must “deliver” something eventually to their clients. But when the perspective on the left dominates,

alternative designs are not considered and the requirements for everyday use may be misunderstood. To a substantial degree, participatory design is spreading to the United States from Scandinavia (Bradley, 1989; Corbett, et al., 1991; Ehn, 1988; Floyd, 1987; Greenbaum and Kyng, 1991), but the understanding of qualitative modeling and the relation of tools to activities and knowledge is not well understood.

Interacting activities when using tools

Implicit in the design perspectives summarized by Table 1 is a fundamental epistemological shift: Not viewing models as equivalent to human knowledge. To understand this, we must understand better how models are used in practice. For example, the research of Shahaf Gal (1991), in collaboration with Donald Schön, examines how designers relate direct physical experience with materials, use of simulation models, and reinterpretation of what representations in the model mean (Figure 1). For instance, in the Janus example presented by Fischer (1991), concepts like “work triangle” in the kitchen redesign computer assistant are related differently to other constraints (and hence given new meaning) by different clients. Rather than seeing a model as replacing encounter with physical stuff in the world or replicating human conceptual structure, we view the combination as a triad: physical being-in-world (creating information by interacting) — (simulation) modeling — conceptualizing. In particular, we must be careful to distinguish using a system like Janus versus standing in a kitchen mock-up. Similarly, we distinguish using Woolf’s cardiac arrest tutor (Woolf and Hall, 1995) and perceiving a patient in the world, moving around spatially, interacting with other people.

The issue is not merely fidelity of isolated information—as this distinction was conceived in earlier ITS work—but the difference between interacting with a program and the choreography of being engaged in multiple activities at one time. For example, a person using a kitchen in the world is not merely “cooking,” but (perhaps) “preparing a meal for a family,” “catering a dinner,” or “carrying on a conversation with friends.” These broader aspects of human activity are nearly always excluded from functional views of places and people, which view action in terms of separated tasks, defined in terms of inputs and outputs. Tasks are either serial or parallel-independent; activities are conceptually coordinated simultaneously. The context of multiple activities and identities makes learning to fulfill a role more than learning scientific facts or operating procedures. In this respect, using simulation tools to replace other sources of information is far too narrow and may distort the context of practice. Effective use of simulations requires understanding how an experienced practitioner relates the constraints of actual practice (and stuff in the world) with different ways of coordinating activity (visually, in gestures,

by description, rhythmically). This is an interactional view of tools—understood as being part of the relation between knowledge and activity, not a replacement or emulator for either.

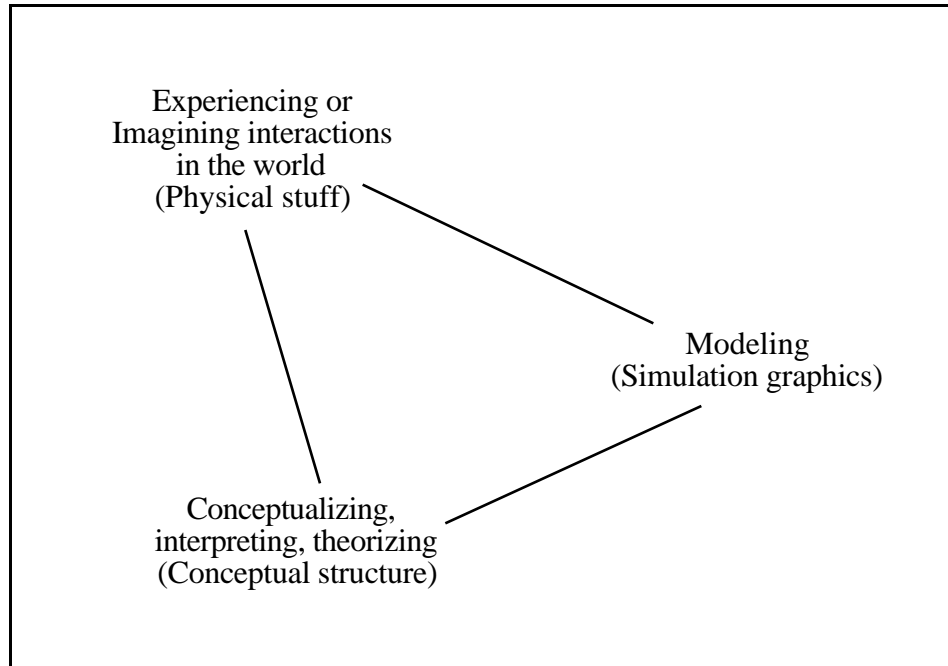


Figure 1: Relating models to conceptual knowledge and activity.

Put another way, models—including especially ITS tools—are appropriately integrated into several interacting activities. In the domain of kitchen design this includes:

- *Projecting* future interactions via imagination and trials with a physical mock-up (includes role-playing and pretending);
- *Modeling* (abstractly representing the situation, such as creating a diagram in Janus); and
- *Conceptualizing* (especially, creating new vocabularies for describing and resolving conflicting constraints, such as understanding the meaning of “work triangle”).

This analysis reveals that a distinction must be drawn between learning about science and learning to design. Design inherently involves participation, values, and human activity. Designing means relating to clients, suppliers, contractors, and the community of practice of other designers—not merely learning principles for good design (Schön, 1987). Tools

for design therefore could extend beyond a “book” view of design to supporting conversations with other people involved in the design or building process.² The same observations hold for troubleshooting, control, auditing, and other engineering and business activities.

The scope of early ITS research limited its value for real-world problem solving. Knowledge was viewed as scientific, objective, and descriptive. The choice of subject material to model reveals this school-oriented, textbook bias: mathematics, physics, and electronic troubleshooting. Even most work on medical diagnosis focuses only on descriptive concepts and presumes that some *individual* (the budding “expert”) is in control. Both the *epistemology* (human knowledge equals descriptive models) and the *organizational perspective* (experts manage tasks) fail to fit the reality of practice. Few ITS systems orient students to other people in the setting who might be involved in framing the problem and assisting. In short, situated learning means interacting within the physical and social context of everyday life, not sitting alone in front of a screen as a disembodied, asocial learner. Again, the point of this analysis is not to rule out simulations of teams (such as an emergency room) or complex physical devices (such as a flight simulator). The point is to understand how a simulation relates to the many ways in which people coordinate their behavior and how rehearsal relates to being on the scene, working every day.

Towards a dynamic view of representational tool creation and use

To understand better how models relate to knowledge and activity, we should consider that models are not merely scientific or derived by scientific experimentation. Models of work include:

- Rules of thumb (perhaps formalized by an expert in a different culture)
- Regulations (global standards, prone to change and/or highly complex)
- Constraints articulated from interactional experience (personal values).

Understanding that professional knowledge is not merely scientific (let alone exclusively descriptive) is part of the realization that learning technology must not only move from the classroom to the workplace, but be based on a reinvestigation of what needs to be

²A major effort to transform classroom learning for children, the MMAP project at IRL, conveys mathematical concepts in the context of design activities (Moschkovich, 1994).

learned. Where do design rationales come from and how are they referred to and used within a community of practice over time? A student is not only learning theories and principles, but new ways of seeing and coordinating action, new ways of talking to other people, and new interpretations of practical constraints.

In effect, instructional designers need a dynamic view of how documents and tools are modified, reinterpreted, and used to create and understand systems in the world. Usually left out of ITS evaluations are the user's experiences and reconceptualizations, which are inherently outside the formalized model of subject material. They are outside because *using* a model inherently means relating it and interpreting it in a dynamic, interactive context of activity, concern, and conflict that the model itself does not represent (Kling, 1991). Even when an ITS is placed on the job, the specification of what it must do leaves out the user's conception of identity and matrix of participation with other people over space and time.

Perhaps the best way to break out of the idea that human knowledge equals descriptive models is to contrast the curriculum orientation of pre- and post-testing and concern with retention with views of learning based on the group's capability. How could an ITS facilitate handling unusual, difficult situations, which are both dynamic and team-oriented? How could an ITS promote organizational innovation and competitiveness? Narrow, individual transfer views of training ignore how ITS might transform organizational learning.

Bringing tools to practice requires additional research and new partners

Understanding new opportunities for learning technology—broadening our view of how ITS tools might be used in practice—we need new partners with new methods for relating designs to practice. In the past, and to some degree appropriately, ITS researchers have adopted the technical perspective of improving the capability of a computer to automate human actions. The two questions asked are generally:

1. How can we represent a design or problem situation? (e.g., layout representation for a kitchen, simulation of a phone menu)
2. How can we provide instructional feedback? (e.g., critiquing and examples)

Research has focused on methods for teaching concepts (descriptions of the world) and what to do, focusing on local constraints. The methods include qualitative modeling, natural language generation and recognition, graphics, and video.

In contrast, a practice perspective—promoting organizational change and customer orientation—would ask different questions:

1. Who can benefit from such a tool? (e.g., experienced designers use the tool for developing theories, instead of *delivering* models to novice users)
2. How can we learn about the appropriateness of the product being designed in context? (e.g., How do Janus users determine their client's needs? How do ITS designers determine that voice dialogues are appropriate for ordering pizzas?)

Research on design in practice has focused on methods for determining how people actually do their work and how their preferences develop through interpersonal interaction. Ironically, like ITS this research is concerned about the “mental models” of users, but it considered instead people's view of how their job relates to the overall organization, their attitude about other people, and their knowledge of other people's capability (Levine and Moreland, 1991). Again, expert knowledge is not just about scientific theory but *about other people*. Ethnographic studies are now applying this perspective to the design of work systems, including organization, technology, and facilities (Kukla, et al., 1992; Scribner and Sachs, 1991).

Helping students understand interactions in practice means relating tools to the context in which they will be used. Fact, law, and procedure views of knowledge tend to leave out this aspect of work—articulating the problem situation, creating new representations in practice. Just as ITS designers leave value judgments out of the knowledge base to be taught, they don't consider how ITS technology itself should be appropriately used. As stated by Alfred Kyle: “We know how to teach people how to build ships, but not how to figure out what ships to build.” (Schön, 1987, p. 11).

Realizing this vision—designing learning technology for practice—requires new research partnerships. Don Norman summarized the process neatly in the title of his position paper, “Collaborative computing: collaboration first, computing second.” The point is that computer scientists need help to understand how models are created and used in practice. The design process must include social scientists who are interested in relating their descriptive study techniques to the constraints of practical design projects:

Building applications which directly impact people is very different from building computational products of the sort taught and studied in standard courses on computer science....

Technology will only succeed if the people and the activities are very well understood....

Computer scientists cannot become social scientists overnight, nor should they....The design of systems for cooperative work requires cooperative design teams, consisting of computer scientists, cognitive and social scientists, and representatives from the user community. (Norman, 1991)

How shall we design the process of ITS design? Probably the simplest starting point in setting up an ITS project is to consider who the stakeholders are and how they will interact with each other during the project. People who might be involved include:

- customers of the company's product or service
- workers, the company's product designers and service providers
- unions
- managers (supervisors, teachers)
- the tool shell and application designers and researchers
- corporate or government "process owners"

Important questions to be negotiated as ways of working together include:

- How are conflicting goals, values, and interpretations represented and reconciled?
- Who is allowed to develop, update, or customize the tool?
- What is the division of labor in system building (teacher, knowledge engineer, programmer) and during instructional interactions?
- How are vocabularies and models shared and reused?

In considering these questions, we must remember that at issue is whether something useful, durable, and scaleable will be developed. When done well, the descriptive modeling aspect of ITS technology promotes a new kind of interplay between theory and practice, requiring people to articulate and reflect on their standards, values, and experience. This interplay is missing from the "delivery" or "capture and disseminate" model of technology development.

As another step, ITS researchers engaged in designing for practice find it useful to report their work in a different way. Rather than only showing representations and subject matter course examples, research presentations comment on generality, the development and formalization processes, and the context of use (illustrated by the articles about Leap

and Sherlock by Bloom and Lesgold in this volume). Each of these considerations is another area for research, changing traditional views about the nature of software engineering and the expertise of computer scientists. Reflecting this shift in concern, claims about instructional tools might be organized as follows:

Applying modeling methods to different domains and tasks

- Does the tool provide a general model of some domain, situation-specific models, or only a calculus for formulating models?
- What assumptions are made about systematicity or completeness of models?

Supporting development and maintenance

- How will authors, teachers, and students learn to use the program?
- What customization is possible for local needs?
- What provision is made for breakdowns in the computer tool itself?

Formalizing authoritative knowledge

- How are local constraints related to global standards?
- Who interprets regulations and policies when creating examples?
- What is the role of corporate training relative to local trainers?

Integrating technology with the context of use

- How is the tool related to existing physical, technical, organizational systems?
- How does learning already successfully occur, without the tool?
- How does the tool relate to ongoing experience on the job?

Such a decomposition of concerns suggests shifting how a tool is presented at conferences and in print: Rather than describing representational techniques alone, developers working in the context of use need to discuss the system that surrounds the tool, including competing methods for learning and the practices that make a tool successful.

Transforming instructional practice: Assessing both technical and social concerns

Many problems arise in developing a computer tool in practice. Until recently, ITS researchers have been preoccupied by technical concerns and have not incorporated methods and partners for handling social concerns. In turn, omitting social concerns

imposes a narrow view of evaluation of ITS tools, which, among other things, inherently devalues the possible implications of this technology on organizational learning. In particular, we might revise our standards of evaluation to assess how ITS technology changes the knowledge of a group.

The questions that are usually raised when evaluating an instructional program involve measurements centered on “courseware,” including authoring time, coverage of the subject matter, media, cost, student time, performance, retention. As mentioned above, issues of generality of the tools and customizability are often raised. A technical perspective leads us to ask, “Should we push the machine’s capability by including natural language input or should we use graphic editors?” “Should we increase the depth of the material or expand the audience?” The technical focus remains on how to get something useful working and then how to make it more technologically flashy, not how to bring about organizational and technological change. But despite the obvious inadequacy of such measures of success, few people question this discourse.

In the 1990s, the widespread availability and use of workstations for clerical tasks has shifted our concern: How might technology be used to help people on the job, such as administrators who cover what were previously multiple jobs in services, technical support, and marketing? Industry has a pressing need for the training of new employees (particularly foreign); key jobs such as telephone service for customers have a high turnover, increasing training requirements. These organizational changes provide an opportunity for reframing how ITS technology is assessed: How might it be designed to support organizational change by supporting learning in everyday contexts?

By relating ITS technology to the current business fad that “one person does it all”—variously called “the integrated process employee,” “the integrated customer services employee,” or “the vertically-loaded, customer-oriented job”—we have the opportunity to jump past the association of computer tools for learning with CBT and classroom instruction. But integrating learning into everyday work requires seeking new forms of assessment and inventing new kinds of supporting technologies for these restructured work functions. The conventional view that solutions to the training problem will “hop from boxes” is obviously inadequate; for the schoolroom, textbook epistemology of ITS is incomplete. Designers and managers alike need to:

—Reconsider what knowledge is required, and then relate the technology to how knowledge develops and standards are articulated and interpreted in everyday conversations between workers.

—Recognize that part of the lack of collaboration between subject matter experts and instructional designers derives from the false expectation that an expert can articulate what he knows; failed collaborations threaten the instructional delivery view that the most important learning occurs by training.

—Recognize that not everyone wants to learn *or will be allowed to*—issues of role, power, authority, and conflicting values must be considered.

The bottom line is that ITS developers must become engaged in a critique of the current fad of “one person does it all” and the “end of specialization,” and help find a middle ground that appropriately reflects what experienced telephone representatives know and how they might productively use computer systems while still interacting with each other.

Such a discussion only begins to address the reality today in relating ITS technology to practice. The reality is a world of downsizing, decentralization, and dismantling training departments (*Training*, 1994). The reality is a world of cost-containment, increasing demands on workers, and increasing reliance on computer systems to store, sort, prioritize, and monitor work. On the plus side, corporate America is becoming aware that the qualitative modeling techniques exist and that computers can be more than automated page turners. The possibilities of the coaching metaphor, multi-agent simulations, conceptual modeling, case-based inquiry, and conversation facilitation are still on the horizon.

But an ITS researcher and designer thrown into this world is faced with a fundamental issue of not just “What ship should I build?” but “How do I change the practice of instructional design?” How do we move an organization forwards? The technologist might have begun by answering, “Give the workers new tools.” But there are an amazing number of alternative and complementary approaches:

- Put people together: encouraging friendship (to promote later collaboration), mimicking, and sharpening identity through contrast
- Dismantle the organization
- Remove control from above; promote bottom-up restructuring
- “Incentivize” people to do something extra (e.g., relate pay to sales)
- Experiment with different designs in different groups
- Send people to seminars
- Politically manipulate or resist the organization

- Quit (join a competitor, start your own company)
- Force change by managerial fiat

In promoting new kinds of learning technology, we must remember that such concurrent influences are part of the process by which new tools are spread, used in new ways, or rejected. Of these, ITS researchers might be advised to use the “put people together approach”: Engage early adopters of the new technology in a project that aims to develop a new understanding of learning and work, involving other participants peripherally, so they can advocate similar projects in their own organizations. In the jargon of organizational change, the issue is to make the methodology “scaleable,” that is, to ensure that it can be understood and usefully employed by other people. Technologists generally view this in terms of “better tools,” but of course managers will assess the methods by cost-effectiveness and demonstrated results.

In summary, developing learning technology in practice—bringing ITS methods to widespread use—involves multiple concerns. The scope of evaluation and perspective on value must broaden from “representing” and “automating” to *changing practice*. The old questions must be juggled with quite different concerns that other players from the social sciences, management, and the workplace will raise and hopefully help resolve. Stated in terms of broadening scope, these questions include how to:

- Create instructional programs with qualitative models and multimedia
- Augment instructional material (e.g., by modeling, graphics, explanation)
- Develop courseware (e.g., help authors, reuse components, facilitate maintenance)
- Evaluate instructional value (retention or satisfy the customer); how observe changed practice and inform instructional designers
- Decide what tools to build
- Integrate tools into existing technological, physical, and social systems which are rapidly changing (when subject matter experts are too busy to help or for tasks that perhaps no one knows how to do)
- Handle political, administrative, and funding conflicts between management, operations, and training departments
- Promote organizational change, innovation, and competitiveness.

Narrow, subject-matter oriented views of evaluation—almost universally adopted by ITS researchers in their attempt to curry the favor of instructional designers—do not

show training to best advantage; they don't show learning on the job as people incorporate and expand beyond the formalized subject material of the curriculum. The ITS community, in attempting to participate in training organizations, has set its sights too low by adopting the individual transfer metaphor and managerial assessments based on what is visible—the number of student hours and the size of binders.

To shift from developing technology to developing organizations, new instructional practice, and a new view of learning, we must reframe the problem around work practice. For example, viewing a community of practice spread over a corporation's offices in several states, we might ask, "How do we support a process at a distance?" We shift from a subject matter view of facts and theories to viewing a coordination problem—scheduling one's time, allocating resources, informing others in a timely manner—the work of choreographing contributions in a distributed system. We shift from delivering someone else's model to asking, "How do we support visualizing, relating and comparing, and organizing alternatives?" We think in terms of tools to help people represent and reflect on their own work, on their social interactions and conceptions about other people.

To do this, developing learning technology in practice, we must involve people in projects that straddle academic disciplines. The best hope is certainly with the next generation of students—in stark contrast with the "knowledge as static repository" view that we must capture and proceduralize the specialized viewpoints of retiring experts. The next generation will learn to describe problems and situations in multiple languages, by different perspectives and methods. This is not the same as delivering a course on one subject.

The problem for people who want to promote this new perspective on knowledge and learning is to find a way to sell what's not in a binder or flashing colorful graphics and sounds in windows. We must develop new forms of assessing learning that make social processes visible and reveal the informal, tacit components of knowledge as essential. The worst possible step, which the ITS community has indeed adopted so far, is to view ITS as supplying training departments with methods to produce more materials or better materials more quickly. Changing the practice of instructional design involves learning about and incorporating new perspectives in computer programs:

- Relating the technology of training to a worker's view of learning;
- Relating the scientific view of knowledge as written facts, rules, and theories to the pragmatics of inventing designs and creatively interpreting policies;

- Relating the individual view of work and knowledge to the social conception of activity, participation, and coordination.

Anyone who has tried to bridge such perspectives knows that the process of change is slow, politically charged, and in many respects uncontrolled. Some researchers will prefer and will be better off working alone in research labs, exploring what computers can do, rather than investigating what people need them to do. Other researchers will find that it is possible to belong to multiple communities, retaining academic ties while enjoying the intellectual breadth of a multidisciplinary team. Ultimately, beyond all the arguments about productivity and competitiveness, it will be individual workers and researchers alike who assess their own performance and realize that in taking on challenges from different perspectives, they have landed on two feet in multiple worlds.

Acknowledgments

This research has been supported in part by Nynex Science and Technology, Inc. and Xerox Corporation, Inc. I would especially like to thank my colleagues for our many stimulating conversations: Gitti Jordan, Charlotte Linde, Ted Kahn at IRL, plus Pat Sachs, Dave Torok, and David Moore at Nynex. Jeff Kelley and others involved with The Productivity Partnership have provided useful observations about trends in corporate training.

References

- Bradley, G. 1989. *Computers and the Psychosocial Work Environment*. London: Taylor and Francis.
- 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. 1987. *Knowledge-Based Tutoring: The GUIDON Program*. Cambridge: MIT Press.
- Clancey, W.J. 1989. Viewing knowledge bases as qualitative models. *IEEE Expert*, (Summer 1989):9-23.
- Clancey, W.J. In preparation. The conceptual, non-descriptive nature of knowledge, situations, and activity. In P. Feltovich, K. Ford, and R. Hoffman (editors), *Expertise in Context*.

W.J. Clancey: Developing learning technology in practice

- Clancey, W.J. In press. A tutorial on situated learning. In J. Self (editor), *Proceedings of the International Conference on Computers and Education (Taiwan)*, Charlottesville, VA: ACE.
- Corbett, J.M., Rasmussen, L.B., and Rauner, F. 1991. *Crossing the Border: The social and Engineering Design of Computer Integrated Manufacturing Systems*. London: Springer-Verlag.
- Ehn, P. 1988. *Work-Oriented Design of Computer Artifacts*, Stockholm: Arbeslivscentrum.
- Fischer, G., Lemke, A.C. Mastaglio, T., and Morch. A.I. 1991. *The role of critiquing in cooperative problem solving*, University of Colorado Technical Report.
- Floyd, C. 1987. Outline of a paradigm shift in software engineering, in Bjerknes, et al., (eds) *Computers and Democracy—A Scandinavian Challenge*, p. 197.
- Gal, S. 1991. *Building Bridges: Design, Learning, and the Role of Computers*. Unpublished PhD Dissertation, APSP Research Program. MIT.
- Greenbaum, J. and Kyng, M. 1991. *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, NJ: Lawrence Erlbaum.
- Kling, R. 1991. Cooperation, coordination and control in computer-supported work. *Communications of the ACM*, **34**(12)83-88.
- Kukla, C.D., Clemens, E.A., Morse, R.S., and Cash, D. 1992. Designing effective systems: A tool approach. In P.S. Adler and T.A. Winograd (editors), *Usability: Turning Technologies into Tools*. New York: Oxford University Press. pp. 41-65.
- Lave, J. and Wenger, E. 1991. *Situated learning: Legitimate peripheral participation*. Cambridge: Cambridge University Press.
- Levine, J.M. and Moreland, R.L. 1991. Culture and socialization in work groups. In Resnick, L.B., Levine, J.M., and Teasley, S.D. (editors), *Perspectives on Socially Shared Cognition*. Washington, D.C.: American Psychological Association.
- Moschkovich, J. 1994. *Assessing students' mathematical activity in the context of design projects: Defining "authentic" assessment practices*. Symposium paper presented at the 1994 Annual Meeting of the American Educational Research Association, New Orleans.
- Nonaka, I. 1991. The knowledge-creating company. *Harvard Business Review*. November-December. 96-104.

W.J. Clancey: Developing learning technology in practice

Norman, D. 1991. Collaborative computing: collaboration first, computing second. *Communications of the ACM*, **34**(12)88-90.

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

Scribner, S. and Sachs, P. 1991. Knowledge acquisition at work. *IEEE Brief*. No. 2. New York: Institute on Education and the Economy, Teachers College, Columbia University. December.

Self, J. 1988. *Artificial Intelligence and Human Learning*. London: Chapman and Hall.

Training, 1994. Re-engineering the training department. *Training*. May, pp. 27-34

Wenger, E. 1990. *Toward a Theory of Cultural Transparency: Elements of a Social Discourse of the Visible and the Invisible*. Unpublished doctoral dissertation, Department of Information and Computer Science, University of California, Irvine.

Woolf, B.P. and Hall, W. 1995. Multimedia pedagogues: Interactive systems for teaching and learning, *IEEE Computer* (Special Issue on Multimedia). May, pp. 74-80.