REPRINT:

Reusser, K. (1993). Tutoring Systems and Pedagogical Theory: Representational Tools for Understanding, Planning, and Reflection in Problem Solving

In Susanne P. Lajoie & Sharon J. Derry (Eds.), *Computers as Cognitive Tools.*
Hillsdale, N.J.: Lawrence Erlbaum Associates, Publishers

5 Tutoring Systems and Pedagogical Theory: Representational Tools for Understanding, Planning, and Reflection in Problem Solving

Kurt Reusser
University of Bern, Switzerland

TUTORING SYSTEMS AND PEDAGOGICAL THEORY

In designing computer-based educational systems our primary concern ought not to be with a dazzling new technology, nor should we be misguided by such romantically unrealistic goals and expectations as replacing teachers, textbooks, or even the physical and social learning activities of students through learner-machine interactions. Instead, the main object in the design of computational media as a new form of "intellectual bootstrapping" (Collins & Brown, 1988) ought to be its functional connection to a (partly normative) pedagogical and didactical[1] philosophy. The design must take into account the proper use and integration of the system into the comprehensive range of learning and teaching activities that take place in the "behavior setting" (Barker, 1978) of *schooling*.

In the first part of this chapter eight principles are suggested for designing computer-based cognitive tools for learning and problem solving. The principles, contrasting in certain ways principles outlined by Anderson, Boyle, Farrell, and Reiser (1984), or by Ohlsson (1986), are based on pedagogical theory and on cognitive research and lead to the discussion of a set of critical issues for the

[1]The concept of didactics is not used here in its narrow and increasingly negative sense of "spoonfeeding" students by extremely teacher-centered instructional methods which leave little of the guidance and learning responsibility to the student, but in its original and broader sense of classical Bildungstheorie (e.g., Klafki, 1963). The latter meaning includes both the question of what to teach (reflecting and constituting the object of instruction) and how to teach (designing an instructional setting of methods and media) a specific content. The two meanings of "didactics" are contained, for example, in the distinction between "product" and "process" with respect to the fundamental pedagogical goals of teaching.

design of educational software. The principles have been applied to HERON, a computerized teaching and learning environment designed to aid children in understanding and solving a wide class of complex mathematical word problems. I describe parts of HERON in the second part of the chapter, emphasizing the important role of *representational means, or formats, as conceptual tools for problem-representation, planning, and reflection.*[2]

From Cognitive Analysis to Instructional Design: Computers as Cognitive Tools

Educational software should make sense from a pedagogical point of view. Hence, four crucial considerations should govern the design of machine-supported instructional contexts: (a) a cognitive and instructionally efficient model of the task or the domain the system is designed for, (b) a sound conception of the general and content-specific learning processes associated with the domain, (c) a domain-appropriate social-cognitive concept of teaching (balancing dimensions such as explicit instruction versus discovery learning, "solo-learning" (Bruner, 1961, 1986) versus collaborative learning), and (d) a view of the active nature of the learner. With regard to the tutoring of mathematical word problems: Whoever designs a computer-based instructional system needs to know both how to effectively represent and convey the informational structures related to word problems and the processes and strategies employed by learners of different ability levels in their understanding and problem solving.

Thus work on tutoring systems should be based both on research in cognitive psychology and on research in didactical or instructional theory, two distinct fields, which still maintain few interconnections. Often enough, cognitive researchers analyze meaning structures and processes on a conceptual level, using formats that are neither translatable into instructionally efficient models of domains and tasks, nor allow inference to any normative principles (Glaser, 1987) of instruction. On the other hand, designers of textbooks and computational media, as well as (expert) teachers, are often not successful in performing microstructural cognitive task analyses, yet such analyses would be beneficial in uncovering the properties of the representational and operative "tacit" (Polanyi, 1966) knowledge inherent in the performance of a task.

This leads to the first principle:

P1: Design and use computer-based tools pedagogically, that is, as cognitive instructional tools for mindful teachers and learners in a culture of problem solving.

[2]Principles are fairly abstract by their nature. The reader who wants to see—while reading the first sections of the chapter—how the principles are incorporated into the tutor, might prefer to read sections of the second part of the chapter first.

In contrast to a technology-driven and opportunistic design philosophy, computers should be used in education, by judicious teachers and active, intentional learners (in the sense of Scardamalia, Bereiter, McLean, Swallow, and Woodruff, 1989) *as supportive cognitive tools in the service of explicit pedagogical goals* (Reusser, 1991). As mind-empowering prosthetic devices which belong to our overall "cultural tool kit" (Bruner, 1990), future computerized tools of learning and instruction not only act as amplifiers of our own intelligence but, beyond that, might significantly change our traditional view of the instructional setting, "redefine the natural limits of human functioning," as Bruner says (1990, p. 21). Instructional tools should be based on our best cognitive analyses of curricular tasks and processes. With regard to their status as supportive mind tools, which are always used in a specific context and with a specific purpose, they should remain a *means to a pedagogical end or goal.* As such they form a functional part of a distributed and much more comprehensive setting for pedagogical intelligence.

From the "Romantic Quest for Intelligent Machines" (Clancey, 1989) to the Activities of Virtually Autonomous Learners

The catchword "intelligent tutoring systems" (Sleeman & Brown, 1982) has come to mean that a computer functions as an intelligent, dynamically adaptive substitute for a human teacher, who is capable of performing sensitive cognitive diagnoses, which means to infer, on the basis of a constantly-retuned student model, a person's cognitive states—what the person knows, how she thinks and learns—on the basis of her overt behavior (cf. Ohlsson, 1986; van Lehn, 1988). There are good reasons to be skeptical about the feasibility—and in part even the desirability—of intelligent systems that are based on full system control and deep student modeling (Nathan, Kintsch, & Young, 1990; Resnick & Johnson, 1988; Scardamalia et al., 1989). Intelligent tutoring, in which a machine tailors its instruction to an individual student on the basis of an inferred, constantly updated, fine-grained mental model, may be seen as a long-term goal. But given the current state of the art, machine-tutoring based on *cognitive simulation* of the student is not possible across a full range of open-ended tasks and domains, where fuzzy language and qualitative world-knowledge based reasoning are required. This is especially true with regard to error modeling. As Derry and Hawkes (1989) note: "Deep modeling of procedural bugs is computationally intractable for complex problem domains, and we do not believe it is required for effective cognitive apprenticeship" (p. 33).

Even if technology based cognitive diagnosis were, in principle, a feasible goal in the distant future, it would be only one of several ways to apply advanced computing technologies to education. From a pedagogical point-of-view, there are alternative means of supporting and facilitating human learning and problem solving through interaction with a computer. The most sensible one might be to view machine-supported tutoring with the ultimate goal of developing a virtually autonomous and reflective learner and thinker. This does not require intelligent systems but flexible, didactic supports. Intelligence, on this view, is seen as being located primarily in the learner and distributed across the whole pedagogical setting, rather than being located in the computer.

Scardamalia et al. (1989) even question how useful highly intelligent systems would be: Such systems

> may also be heading in the wrong direction. For it is not the computer that should be doing the diagnosing, the goal-setting and the planning, it is the student. The computer environment should not be providing the knowledge and intelligence to guide learning, it should be providing the facilitating structures and tools that enable students to make maximum use of their own intelligence and knowledge. (p. 54)

One should therefore not conceive of computer applications in education primarily as substitutes for intelligent teachers but as tools aimed at cultivating the intelligence of the user, as didactic instruments directed, to the greatest possible extent, at fostering learner autonomy and self-regulation. One can add that there is little evidence that even expert human teachers are able to carry out extensive cognitive diagnosis (e.g., McArthur, Stasz & Zmuidzinas, 1990).

P2: Extend and empower the minds of intentional learners.

Computer environments should be seen as mind-extending or catalyzing tools for intelligent and volitional learners and virtually autonomous problem solvers. They should provide stimulating and facilitating structures in order to promote meaning construction activities, such as planning, representation, and reflection. Such an alternative view of computers can be situated within the epistemological and didactic framework of models and metaphors currently being discussed in applied metacognitive research. These include, for example, the Vygotskian-inspired models of coaching and scaffolding (Brown & Palincsar, 1989), of cognitive apprenticeship (Collins, Brown, & Newman, 1989), of procedural facilitation (Scardamalia et al., 1989), of learning through reflection (Collins & Brown, 1988), or, more generally, of autonomous and self-directed learning and problem solving (Beck, 1989; Bruner, 1986).

From this pedagogical perspective, tutoring would not be considered successful primarily with respect to the degree to which a system is able to "intelligently" force a student down some preset solution path, as human tutors very often do but, instead, to the degree to which it optimizes students sense of control, and to the degree to which student solutions are self-generated.

P3: Provide learners with some guidance according to the "principle of minimal help." (Aebli, 1961)

Making errors, or getting stuck, is an inevitable part of learning. However, in order to provide effective feedback or graduated help, a tutor does not necessarily need to perceive what the student is thinking but to know what the structure of the task is, and what the student is doing while working on it. Since learners can become highly confused and demoralized by undetected errors (Anderson et al., 1984), some feedback must be provided—either immediate, delayed, or on request. Good teachers, as well as intelligent learners, follow the didactic "principle of minimal help" (Aebli, 1961). Ideally, a tutoring system would leave it to the student, to use or seek only as much help or feedback from the system as he needs. If tutorial action appropriate for an individual learner is called for, cognitive modelling, however, is not the only way to determine its quality. An alternative basis for characterizing students' errors and making tutorial decisions can be established through a different form of behavioral diagnosis (cf. Wenger, 1987). It requires a careful conceptual analysis or decomposition of the knowledge or skill to be taught. The tutor should know mature (expert) and less mature models of the processes and representations to be taught. Feedback during problem-solving, for example, on errors, can be based on a conceptual analysis of the solution space. This makes it possible to determine when and how the observed knowledge-construction activity of a particular student deviates from a predetermined set of solution paths (Derry & Hawkes, 1989). Thus, mapping overt student performance onto powerful representations of a task can lead to effective guidance without assessing student thinking on a moment-by-moment basis. Still another type of feedback is used by Nathan, Kintsch, and Young (1990), in their tutoring of distance-rate problems. The student can run an animation (time varying computer graphics) on the basis of his problem model, enabling him to judge its correctness on his own.

From Learning through Memorization and Drill-and-Practice Routines to Learning through Active Construction, Comprehension and Reflection

Computer-based cognitive tools should be less oriented toward memorization and drill-and-practice and, instead, toward fostering meaning construction activities, like understanding, problem solving, planning, and reflection. Computers with today's direct-manipulation graphic interfaces (Hutchins, Hollan, & Norman, 1985) are best equipped to do (and undo) such things as generating icons, selecting, presenting, touching, linking, placing, storing, and retrieving information, including bookkeeping and monitoring of the user's actions. Thus they are ideally suited to providing both representational and procedural facilitation to the student's understanding.

P4: Have students construct and externalize their mental models.

Uncovering the covert, or externalizing the hidden, intermediate and component steps and products of the learner's "effort after meaning"—as Bartlett (1932) who was a constructivist far ahead of his time, framed the self-constructive nature of the creation of meaning—is a major cognitive function of computer-based instructional tools. Normally unobservable knowledge-construction activities that are reified (Collins & Brown, 1988) as accessible visual displays reduce the burden on working memory. What has been externally represented, objectified, embodied, organized, and made overt and explicit by *extracortical organizers of thought* (Vygotsky, 1978), can then be identified, inspected, analyzed, discussed, communicated, further reflected and operated upon, and finally carried out consciously and deliberately by the learner (Greeno, 1987; Pea, 1987).

By proposing that students make their thinking explicit and that they actively construct their own conceptualization of a problem or domain, we do not pretend any quasi-automatic and significant improvement of knowledge organization or higher-order thinking skills. Just as an empty head cannot think, there is no effective computer-supported learning without domain-related, representational and procedural (strategic) tools supplied by the educational culture. This claim is in direct opposition to the empirically unwarranted, romantic growth optimism currently in vogue, an optimism that is inspired by Piagetian ideas of cognitive growth and maturation (cf. Aebli, 1978), by related ideas of radical constructivism and discovery learning, as well as by a superficial understanding of concepts proposed by Vygotsky. Proponents of this view sometimes seem to believe that the self-construction abilities of children, as well as skill and knowledge formation in general are simply emergent, nonintentional properties of mostly nondirective or nonauthoritarian social interactions between children and more knowledgeable others.

P5: Provide students with intelligible and effective representational tools of thought and of communication.

Efficient conceptual representation of content is a key problem for both learning and teaching. Appropriate representational formats of domains and tasks, including tree structures, coordinate graphs, diagrams, data tables, conceptual networks, symbol systems (alphanumeric, algebraic . . .), and scientific notations, are indispensible tools not only for thinking, problem solving, and reasoning, but also for the *communication of knowledge*. Tutors and textbooks should provide students, as an important target of instruction, with cognitively plausible operative, iconic, and symbolic systems of representation rooted in a deep semantic understanding of the domain.

Finding facilitating representations for almost any (class of) problem(s) should be seen as a major intellectual achievement, one that is often greatly underestimated as a significant part of both problem-solving efforts in science (Simon, 1977)[3] and of efforts in instructional design. Teachers and designers of knowledge media, should take pedagogical responsibility for giving students the power of effective *domain ontologies* (Greeno, 1983). That is, they should supply students with carefully designed conceptualizations, symbol systems and instructional models of tasks and concepts, that are vital for the development of expertise in almost any knowledge domain.

But what are cognitively plausible and efficient systems or forms of representation? If supplying carefully designed conceptualizations of problems and domains for students is a significant cognitive issue and a major pedagogical device, questions arise about how to characterize the qualities of *good* instructional representations. There are at least two related issues involving *goodness:* One is the question of representations as *domain ontologies* (Greeno, 1983; Wenger, 1987), also called the issue of *cognitive and epistemic plausibility or fidelity* (How *faithful* is a conceptual model of a domain and what are its *ontological commitments?*). The second is the issue of representations as *pedagogical means* of looking at a topic or domain (tools of *Anschauung*), an issue, as Ohlsson (1987) remarks with respect to a proposed "pedagogy of illustrations," that is not yet well understood.

I think an answer to the question of the cognitive-didactical quality of instructional representations contains several elements. They are outlined next and are incorporated into the tutoring system[4] described later in this chapter:

1. Cognitively plausible and pedagogically useful representational systems or formats allow students, while creating and elaborating a mental representation of a problem, to capture (the) essential structural features of the problem and to differentiate the problem from classes of similar problems. Efficient representations permit one to organize a task around salient properties and invariants of its (functional or relational) deep structure, i.e., around abstract relations among components—something that, for example, experienced teachers and expert problem-solvers do intuitively. It requires breaking up a domain into *conceptual building blocks* in such a way that the natural and conceptual con-

[3]Scientists, especially mathematicians and logicians, always have devoted much energy to the development of useful and efficient forms of symbolization. As Simon (1977) notes, there might be only a few basic formats of representation at all in science. On the psychological level of the individuum, numerous studies show the superiority of experts in knowledge organization and problem conceptualization: Skilled problem solvers not only build rich problem representations before they start solving a problem, but good representations of problems significantly affect problem solving efficiency as well.

[4]The basic representational format and strategy of our system are solution trees. The reader may read the sections on this format in parallel with the following list of components of efficient representational systems.

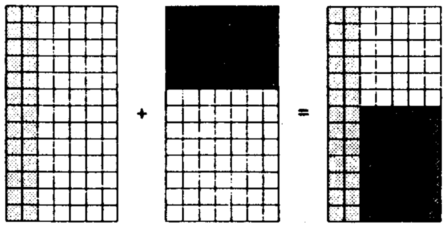


FIG. 5.1. Instructional representation designed to provide an iconic understanding the operative structure of fractions addition. The task is

$$\frac{2}{7}+\frac{5}{13}=\frac{2\times 13+7\times 5}{7\times 13}$$

From Van Hiele (1986). Reproduced by permission of Academic Press.

straints inherent in the domain or task become explicit and easier for the learner to grasp. In a sufficient and concise representation, everything that is needed for processing is contained in it, and everthing that is contained, is relevant.

2. *Representational notations guide students' problem-solving and knowledge-construction activities by supplying operative, iconic, and symbolic forms of solutions and—more generally—of understanding.* Efficient representational formats make evident the constraints that problem spaces and solutions must satisfy. Well-defined notational formats can serve as a form of *written calculus* (Clancey, quoted from Wenger, 1987, p. 320) for a domain or a class of problems. They reflect the constraints of a task, direct students' construction of mental models in ways theoretical task analysis says are required for solving, and allow "the quality of solutions to be evaluated" with respect to its form (Wenger, 1987, p. 320).

3. *Good representational formats enforce intentional structural editing, that is, they encourage students to view their manipulations of a representation as semantically meaningful operations.* This can be encouraged (a) by supporting different ways of conceptualizing, or multiple solution paths, (b) by allowing the student to reconfigure a construction process or to refer back to prior parts of it, and (c) by discouraging referentially empty manipulations of the mere syntax of a representational format.

4. *Effective representations allow rapid recognition and retrieval of relevant information, mainly by reducing abstract problem-solving and reasoning processes in favour of processes which come closer to perceptual operations, to seeing things* (cf. Wertheimer, 1945). The great utility of computationally efficient diagrams "arises from perceptual enhancement" (Larkin & Simon, 1987, p. 95), the fact that relevant relations and conclusions can be easily computed and read off with the help of "simple, direct perceptual operations" (Reusser, 1984). Two diverse examples of efficient iconic representations are depicted in Figs. 5.1 and 5.2. Further examples include the coordinate system, or function diagrams.

5. *Effective representational systems provide a structural basis (platform) upon which, using domain-specific or general problem-solving methods or strat-*

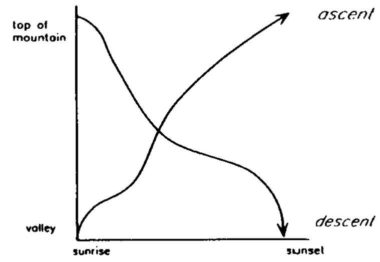


FIG. 5.2. From Duncker (1935) stems the following problem: One morning at the time of sunrise, a monk started out to climb a high mountain. A narrow spiral path led upwards to a temple. After some days of Lent and of meditation, again at sunrise, the monk came back down to the valley following the same path downwards that he came up. Does there exist a location on the mountain path which the monk reaches at exactly the same hour of the day on his ascent and on his descent?—The visual diagram maps in a productive and unique way the problem situation onto the standard notational format of the coordinate system. To most readers, the visual representation elicits a kind of a gestalt switch, of sudden restructuring.—The problem is easily solvable if one succeeds in representing it (either externally or mentally) in a visually insightful way, for example, by assuming that two monks do the ascent and the descent on the same day. The given pictorial representation allows one to read off, with simple perceptual operations, the solution that the monks must meet each other at some place.

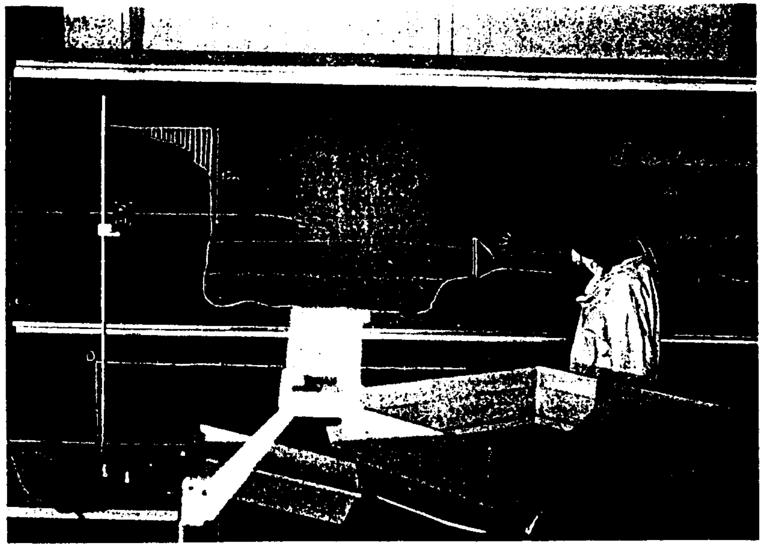
egies, the user can act, manipulate, and reason. Their efficiency also arises thus from *operative enhancement*, the fact that procedural formats of representation support certain kinds of actions and transformations on its objects. Examples include the widely used and fundamental procedural formats of alphanumeric and algebraic equations.

6. *Instructionally valuable representations serve to mediate between idiosyncratic and informal analyses of problems and concepts and shared cultural and more formal analyses.* Among their most important dual function is to provide bridges from ordinary language, or the physical view of concrete objects, to canonical scientific languages and conceptualizations. That is, efficient representations inherit a dual role: They should be linkable, on the one hand, to common natural language descriptions of reality and to informal analyses of problems and domains, including learners' everyday mental models and intuitions.[5] On the other hand, they also contain conceptual elements that correspond to variables in abstract views, scientific models and formalisms, as, for example, mathematical notations.

7. *Hence, cognitively plausible instructional representations should be parts of learning systems* (Nesher, 1989) *in which multiple representations, designed*

[5]Nesher (1989) makes an important point to this in saying that artificial representations should enrich, and not simply replace "the child's intuitions from his everyday experience" (p. 212). As a general principle, theoretical knowledge that can not be deeply connected with the learner's old concepts and intuitive theories, does not become instrumental or integrated into his existing knowledge base. On the contrary, it remains alien to it.

to preserve different aspects of an invariant relational structure, are linked in a yoked fashion (Resnick & Johnson, 1988). Choosing "an apt combination of situational and quantitative models for instructional purposes is a challenging problem" (Hall, Kibler, Wenger, & Truxaw, 1989, p. 280). Connected representations allow the student to view the same object, relationship, or process, from different representational perspectives. By fostering *mobility* between multiple representations (Aebli, 1981; Bruner, Olver, & Greenfield, 1966; Piaget, 1947), it should become clear to the learner "that it is not the representation on the screen that matters in the end, but the representation built up in the students'



FIGS. 5.3a–d. Teaching the law of falling bodies to 14 year old Swiss eighth-graders with the aid of a yoked (discovery learning) system of representations (after Hollenstein, Staub, & Stüssi, 1987). The system was successfully used in classroom instruction and by groups of students. At the beginning, the following problem was posed to the students: A stuntman wants to jump with a motorcycle a gorge 50 meters wide with a drop of 15 meters. What must his initial speed be?
FIG. 5.3a. In the classroom, a simulation of the problem situation was built up and represented with the aid of iconic, manipulative, and symbolic elements, that is: a sketch including the gorge, the ramps for the jump, and the ideal trajectory (1) was drawn on the blackboard; in order to simulate the possible effects of the varying speed of the motorcycle on the trajectory, a jet of water (2), adjustable in pressure, was projected on the wall (3); a computer graph depicting the coordinate system, allowing to study the jet of the water as time-varying graphics, could also be projected.

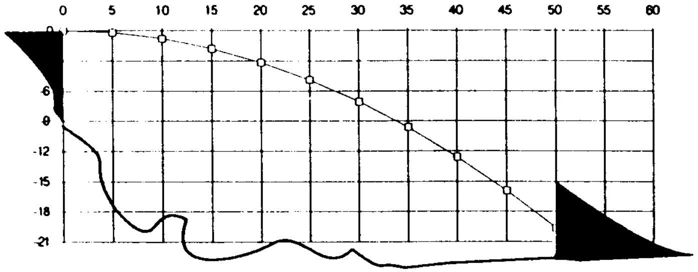


FIG. 5.3b. The figure intimately connects to Fig. 5.3a. It represents the problem situation in a more abstract way. The representation is generated by a personal computer (where the problem is implemented with graphic and symbolic tools) which can be used by the students for setting up theoretical experiments and for mathematization.

head—their mental models" (Resnick & Johnson, 1988, p. 27). Figure 5.3 shows parts of an connected learning system of multiple, progressively more abstract instructional representations designed for teaching the law of falling bodies, starting from study of the trajectory (parabola) of horizontally thrown objects.

8. *Externalized representations supply teachers and students with a conceptual language to communicate and talk about what is to be learned.* They give referential meaning to students' thinking and discussions of the task, but also to the instructional dialogue between teachers and learners.

Representation has been a perennial issue in problem solving literature since Gestaltpsychology (Wertheimer, 1945). As a classical wisdom in problem solv-

FIG. 5.3c. Representation of the underlying mathematical problem structure as computer-generated data tables (Columns: left = falling time; middle = horizontal distance; right = vertical distance) according to parameters that were set by the students. Comparing different tables allow the students to inductively infer the law of falling bodies: the handwritten entries refer to the comparison of the change in (falling) time with the change of the horizontal and vertical speed.

velocity	72	km/h	20	m/sec
time interval	0 5	sec		
start with	0	sec		
time (sec)		horizontal (m)		vertical (m)
0		0		0.000
0.5		10.000		-1.226
1		20.000		-4.905
1 5		30 000		-11.036
2		40 000		-19.620
2 5		50 000		-30.656
3		60 000		-44 145
3 5		70 000		-60.086
4		80.000		-78.480
4 5		90.000		-99.326
5		100 000		-122.625
5 5		110.000		-148.376

t (sec)	d: depth of drop (meters)	calculation of d
1	4.91 m	1 * 1 * 4.91
2	19.62 m	2 * 2 * 4.91
3	44.15 m	3 * 3 * 4.91
4	78.48 m	4 * 4 * 4.91
.	.	.
.	.	.
.	.	.
t	d	t * t * g/2

FIG. 5.3d. Abstractively reduced data table from Fig. 5.3c generated on the blackboard during classroom discussion.

ing says: to properly understand a problem is halfway to the solution. To solve a problem means first to *understand* it, to represent, or see its inherent structure, which means to build an appropriate internal and/or external conceptualization or rich data structure. Most problem-solving processes inherently consist of a representation-construction part which is followed by problem-solving operations that act upon the created representation. Thus, there can be no doubt that carefully designed instructional models of tasks and domains, which facilitate the organization and (re)construction of meaning in knowledge acquisition and problem solving, constitute vital tools and targets of both learning and teaching—both within and outside of computer-assisted instruction.

However, the design of plausible and efficient representations as *instruments of thought and communication* (Kaput, 1989) is more than a prerequisite pedagogical task, and far more than just ad hoc and tricky didactic art work to be quickly replaced by canonical (symbolic) notations and standard conceptualizations of science. The issues of *computational efficiency* (Larkin & Simon, 1987;

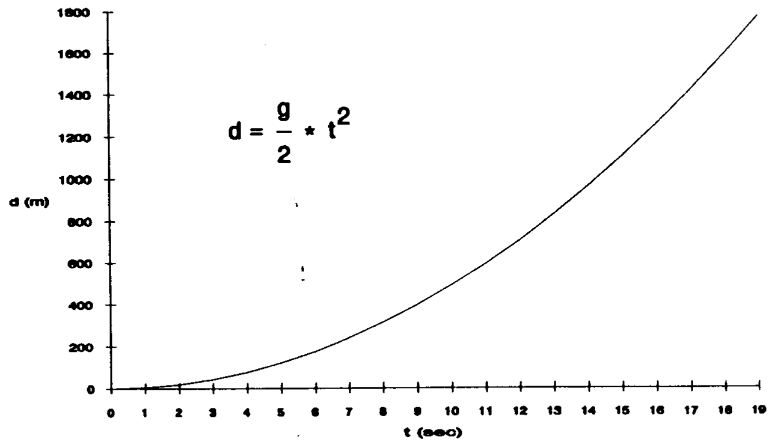


FIG. 5.3e. Function graph and corresponding algebraic equation for the law of falling bodies.

Winston, 1984), of *conceptual faithfulness* and *cognitive and epistemic fidelity (Roschelle, 1990; Wenger, 1987)*, or of *intelligibility (Reusser, 1984)* of effective instructional representations touches the fundamental epistemological question of what are the building blocks, the psychological forms or mediums of knowing and thinking (Aebli, 1981), and the *conceptual entities* (Greeno, 1983), out of which our most effective mental models and our most satisfying cognitive activities are made. Thus, there are questions of ontological and epistemological commitments and implications which are not beyond pedagogical design but intimately and inseparably connected to it.

P6. Promote the use of comprehension-related strategies. Together with representational formats, general and domain-specific strategies are the cognitive tools of thinking and problem solving.

Complementary to providing facilitating representational entities and formats should be procedural assistance, the facilitation of domain-specific and global strategies. While advice with respect to high-level (meta)cognitive (control) strategies like planning, reflecting, setting and maintaining attention to goals, searching (for alternative solution paths), or monitoring one's own performance, is easier to provide from computers in many cases (cf. Cumming & Self, 1990)—and also seems to be transferable to other domains (Brown & Campione, 1990), sensitive task-specific help is a far more difficult problem. Domain-specific *strategic* assistance, that is, assistance beyond simply providing solutions via "informed feedback" (Wenger, 1987), can only go as far as it is possible to formalize the semantics of a domain. And, as a corrolary to this, it will be successful only in so far as cognitive or behavioral diagnosis are feasible.

P7. Encourage reflective and self-directed learning.

Pedagogically designed computer environments, which at the same time enable and force students to uncover and reify their knowledge-building activities, provide a motivating and powerful medium for self-paced reviewing, discussing, and reflecting upon one's own thought processes. Collins and Brown (1988) have framed the notion of the computer as a "tool for learning through reflection." This refers to the unique power of the computational medium to "keep track of the actions used to carry out a task" (p. 1), to display thinking paths, and to allow students to focus and reflect on the why's and how's of their own problem solving—all at their individual pace and according to their own direction. Giving students over and over again opportunities to monitor on-line the visually displayed traces of their planning and thought processes, including alternative routes taken through problem spaces, and to retrospectively analyze those traces and products by reconsidering what has been done, may eventually lead—beyond the acquisition of domain-relevant strategies and control structures—to an overall *reflectivity* which characterizes more mature and expert learners and problem solvers.

From Robinson-Crusoe-Learning[6] to Supportive Contexts of Collaborative Learning

Intelligence should not be seen as a property of the mind alone but rather as a quality that is distributed among the components of learning systems and the social-cognitive environments in which they are embedded (Bruner, 1986; Pea, 1987; Salomon, Perkins, & Globerson, 1991). Apprentices and nascent experts of almost any demanding domain don't develop professional knowledge and skill like the lonely Robinson, that is, as single and independent learners. Instead, they receive substantial expert guidance in instructional settings which supply rich knowledge sources and competent scaffolding. A significant part of learning occurs in interaction with more knowledgeable and skilled, significant others (Mead, 1934; Vygotsky, 1978). It is, in contrast, a certainly questionable feature of our traditional culture of schooling that students are treated almost exclusively as lonely, single learners—as *solo learners*, as Bruner (1986) says. It is unlikely that in the near future computers will become really good conversational partners or sensitive coaches and critics. However, as components that help foster cooperative learning, they can play an important role in classroom learning where collaborative work is supported.

P8. Extend the use of computer-based instructional tools into a supportive classroom culture of collaborative learning.

Computers should not be seen primarily as isolated tools for single learners but rather as instructional devices in classroom environments that support collaborative learning. Computers permit teachers to arrange a broad variety of collaborative learning activities around the reified conceptualizations of students:

- Small learning groups can look back over their comprehension or solution paths and mutually discuss their situation models;
- learning dyads and groups might also view (abstracted) replays or animated traces of solution paths (including those of experts), interrogate aspects that were different, and reflect by which changes they could be improved (Collins & Brown, 1988; Lajoie, this volume);

[6]Robinson Crusoe, the hero of Daniel Defoe's (1720) famous adventure story, who is cast up on a lonely island and condemned to reinvent and rediscover the tools of culture on his own, is the prototype of the lonely individual learner. Another romantic prototype of lonely learning is Jean Jacques Rousseau's (1762) solo learner Emile.

- study partners can collaborate in ways similar to patterns of cognitive apprenticeship, where partners alternatively assume the job of monitoring processes or of scaffolding a set of strategies;
- finally, pairs of students can engage in a dialogue while jointly planning a solution or constructing a shared situation model, thereby developing, refining, tuning, or repairing each other's mental models of the task (Roschelle, 1990; Salomon, 1990.)

HERON: A COGNITIVE TOOL FOR UNDERSTANDING AND SOLVING COMPLEX MATHEMATICAL WORD PROBLEMS

In the second part of the chapter, I describe a computer-assisted learning system called HERON,[7] developed for the domain of mathematical word or story problem solving, which was designed around the pedagogical and cognitive principles outlined above. I begin with a brief overview of the system and then describe the cognitive-pedagogical analysis of the task. In so doing I concentrate on solution trees as the representational format used for instruction in planning and problem conceptualization in this domain. Finally, I describe in some detail an example of how it can be used for instruction.

As is known from educational practice and from countless studies conducted over the past decade (for a review, see Verschaffel, in press), mathematical word problems are difficult for students at all grade levels. At critical points of students' school careers, applied mathematical problems are often used to assess situated mathematical knowledge and cognitive skills, such as planning, problem-solving, reasoning, or abstraction.

Mathematical word problems contain a description of an action, story, or process structure and an implicit or latent mathematical structure. Both textual worlds, which are interwoven with each other, are related by a problem question defining a variable, the value of which has to be determined.

The system HERON was designed to assist children to improve on this curricular task. It is related to three main lines of work: to the theory of discourse comprehension of van Dijk and Kintsch (1983), to our cognitive simulation work on problem comprehension and mathematization (Reusser, 1985, 1989a, 1990a), and to the cognitive and instructional framework of Aebli (1980, 1981, 1983), including work on solving complex mathematical story problems (Aebli & Staub, 1985; Aebli, Ruthemann, & Staub, 1986).

HERON is a *graphics-based instructional tool for facilitating and fostering*

[7]After the Greek mathematician HERON of Alexandria (appr. 100 BC) who created some of the first mathematical word problems, still found in modern mathematical text books, and who, in his book *De automatis*, far ahead of our time thought about the facilitation of life by building machines.

self-directed understanding and solving of complex mathematical story problems. The system is designed to assist students from grade levels 3 through 9 in understanding the language of a problem in order to construct internally a concrete episodic situation model, and to construct externally an explicit and reified mathematical problem model from which a linear equation can be derived.

The system provides the user with a basic instructional format or tool for problem representation—tree structures—and two kinds of instructional strategies that can be accessed. *Solution trees* or *planning trees* are used by the system to conceptualize mathematical problems at an intermediary, bridging stage between text-surface and underlying equation. The first kind of strategy is directed at text comprehension: the analysis of the problem-text in order to build an internal model of the problem situation (situation model). These relatively weak strategies, which the user can call on as needed, supply explanatory help with respect to the vocabulary, the syntactic and the semantic structure of complex texts. The second kind of strategy aid is directed at the constructive abstraction of the mathematical structure of a problem. It helps students to identify, analyze, and conceptualize the relevant pieces of information, and supports the reified planning and construction of a mathematical problem model, including the derivation of an equation.

HERON gives the student a fair amount of interactive flexibility and a high degree of control both in conceptualizing the problem and in planning the mathematical solution. Although HERON does provide instructional help according to students' needs, it does not perform any behavioral diagnosis on the basis of *student modeling*. Instead, our approach follows the design philosophy of *unintelligent tutoring* put forth in ANIMATE by Nathan, Kintsch, and Young (1990). However, in contrast to ANIMATE, which does not try to understand the student at all, HERON performs a behavioral analysis of what the student is doing on the basis of cognitive task analysis.

Cognitive Task Analysis I: From Text to Situation to Equation

Professional teaching should be based on a sound cognitive psychological and didactic decomposition of the curricular task at hand, including an analysis of the product and of the processes involved. As a starting point for teaching children to understand and solve mathematical word problems, one needs first a clear picture not only of the mathematical and the domain concepts involved, but also of the underlying comprehension and mathematization skills by which students of various levels of ability and practice extract mathematical information from verbal problem statements.

Understanding mathematical text problems is a complex and knowledge-intensive inferential and highly constructive process that requires skillfull interaction of more than one kind of knowledge, including linguistic, situational, as well as mathematical knowledge. In school mathematics, this interaction entails transforming natural-language problem texts into some canonical form of mathematical expression, for example, an equation. In this process, the mathematical deep structure of a word problem is merely *one* constraining factor for getting at the right arithmetic strategies. Indeed, factors other than mathematical skill are a major source of difficulty with word problems (Cummins, Kintsch, Reusser, & Weimer, 1988; Staub & Reusser, 1991). Thus, linguistically cued situational and mathematical understanding is not optional, or superfluous, but a helpful and mandatory achievement (for empirical evidence see, e.g., De Corte, Verschaffel, & de Win, 1985; Hudson, 1983; Reusser, 1988, 1989b).

By integrating work from Aebli (1980) and from Kintsch and Greeno (1985), I have developed a rule-based simulation model (Reusser, 1985, 1990a) that illuminates the role of language and situational factors in understanding and solving word problems, and that provides explicit and detailed descriptions about the tacit knowledge involved in these processes. The computational model takes elementary addition and subtraction word problems as input, understands and solves them by using various strategies (in the sense of van Dijk & Kintsch, 1983) and by creating several transient representations based on the words in the problem texts. The process includes the construction of four interrelated and mutually constraining mental representations or levels of comprehension: a *textbase* as a propositional representation of the textual input, a *situational model* as an elaborated qualitative representation of what the text is about, a *mathematical problem model* as the abstract gist of the situation, and an *equation* (Fig. 5.4).

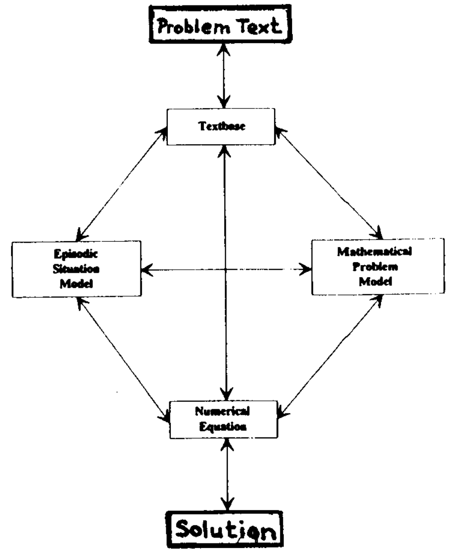


FIG. 5.4. Four interconnected levels of comprehension in solving mathematical word problems (after Reusser, 1985).

The ideal didactic sequence for learning the right *mapping between text, situation model, and equation,* is to consciously go through and elaborate these levels of comprehension. But it will be no surprise that students often deviate in characteristic ways from this pattern. There are the skilled students, on the one hand, who frequently skip a detailed qualitative modeling of the structure of the situation. Their mathematical strategies, deeply connected with rich patterns of situational knowledge, allow them to get directly at an abstract mathematical problem model or the numerical equation. When reading a problem they can almost immediately "see" the right mathematical schema, trigger a smart super-strategy and map it onto the problem situation. Weaker or novice students, on the other hand, who adopt similar strategies of "direct translation" (see Bobrow, 1964) by trying to jump from the text to an equation, fail with the same tasks. Still other students, who are severely lacking in all types of relevant knowledge and skills (linguistic comprehension skills, domain knowledge, and efficient mathematical knowledge) sometimes adopt even worse *coping* strategies that bypass the logic of mathematical sense-making activities (Lehtinen, 1989; Reusser, 1984, 1988; Schoenfeld, 1989). For example, there are students who simply plug numbers into some equations, or perform various kinds of "magic" number work. These students need the guidance of an adequate pedagogical setting. They have to learn to analyse and conceptualize a problem step by step:

- First, they need help putting the problem into a language that allows them to connect its semantic content with their everyday and intuitive concepts and experience;
- next, they must work towards constructing a mathematical understanding of the problem (by establishing the appropriate intermediary problem representations;
- finally, they must map these onto a formalized notation in a canonical format.

Cognitive Task Analysis II: Solution Trees as Representational Tools

Conceptual representation, which is related to the crucial issues of *task analysis* and of *problem space reification,* has been the major driving force for the design of HERON. Although ordinary language provides us with the single most important medium (Bruner et al., 1966) for communication and representation of meaning, there are certain kinds of information that cannot be adequately expressed linguistically. Because it lacks explicitness and because it contains many irrelevant details, ordinary language is not an efficient instructional representation for mathematical structure, for example, for the quantitative entities and relations implied by mathematical story problems. The mathematization of a word problem requires a step-by-step transformation of its textual structure into more adequate, perceptually enhanced and computationally efficient (Larkin & Simon, 1987) forms of representation, ultimately a numeric format. A crucial step in this process is to find *auxiliary representations* (Paige & Simon, 1966) which are intermediary to both the textbase (Kintsch, 1974) and the underlying mathematical structure, that is, which mediate between language and situation comprehension and quantitative, or mathematical thinking.

HERON uses a graphical format for problem representation, planning, and reflection, called *calculation or solution trees* (Aebli, Ruthemann, & Staub, 1986; Derry & Hawkes, 1989),[8] or *conceptual planning trees.* As an analytic tool, solution trees supply a network formalism of dynamically linked entities designed to capture the operative, semantic-mathematical deep structures implied in a broad range of story (algebra) problems. As a mental modeling tool for students, the tree structures, which can be flexibly manipulated and visually inspected, provide a means for reifying both planning and construction processes and the (intermediary) products of understanding.

The semantic building blocks of solution trees are *domain-specific relational schemata,* each schema forming a subgoal in an arbitrary complex, hierarchical solution tree. Each relational schema or triad consists of a pair of qualitatively and numerically specified *situation units,* allowing the computation of a third, unknown unit. Each situation until is expressed as a box containing three subfields of information: a field for the *numerical value,* which may be unknown, a field for the *unit of measurement,* and a textual label field, containing semantic information about the unit's *situational role* which links the quantitative information to a qualitative situation model.

Examples of the graphical form of the situational elements composing different relational schemata are depicted in Fig. 5.5. Domain-specific relations—*Sachverhältnisse,* as Selz (1922) called them—form the basic semantic units of HERON. Their conceptual complexity is a major source of the difficulty of (algebra) word problems, requiring situated, world-knowledge-based reasoning.

Solution trees in HERON are constructed through mixed forward- or backward-inferencing activity. Forward-inferencing means that the student's solution starts by constructing triads based on the quantities given in the problem statements. Backward-inferencing means that the student starts with the goal element contained in an explicitly stated or inferred problem question and works backwards to the given elements via some intermediate calculational levels. In any case, quantitative elements of the problem text, which are recognized as solution-relevant, problem-specific mathematical relations, are combined first with local, then with progressively larger, hierarchical compositions of triadic schemata. The (sub)goal-driven construction proceeds until the mathematical solu-

[8] Derry and Hawkes (1989) use an almost identical concept of solution trees as a basic representational format in their system, TAPS. Both groups have developed their ideas completely independently of each other.

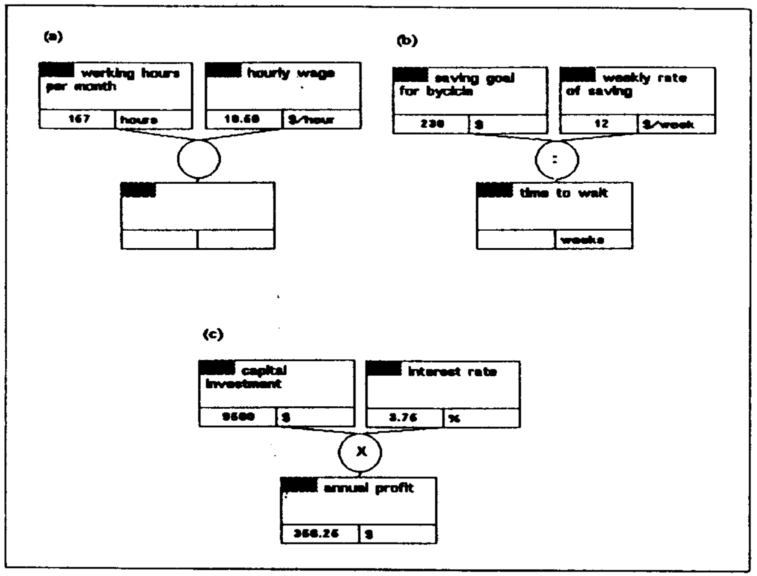


FIG. 5.5. Three nascent versions of domain-specific relational schemata or Sachverhältnisse (Selz, 1922) as examples of the building blocks of solution trees. (a) and (b) are incomplete, (c) is completed. Each schema relates (R) two semantic concepts in a mathematically meaningful way. Other examples: R (distance, speed, time); R (initial price, discount, sale price); R (volume, mass, density).

tion, constrained by the form of the completed planning-tree, can be computed from a final triadic schema.

Figure 5.6 shows two solution trees describing alternative mathematical conceptualizations of the same problem including different final equations. Both solution paths can be mapped onto each other through the laws of associativity and commutativity.

Taken together, there are many reasons to consider solution trees (ST) as cognitively plausible and useful representational and conceptual tools in HERON and as a goal for instruction. Solution trees incorporate most of the qualitative features outlined earlier (see principle P5):

- they are transparent, self-explanatory, and visually inspectable cognitive instruments for representing, evaluating, and communicating the processes of understanding and solving of a large class of word problems;
- they are manipulable, perceptual, highly dynamic and flexible forms for constraining knowledge-construction processes and their (intermediary) products;

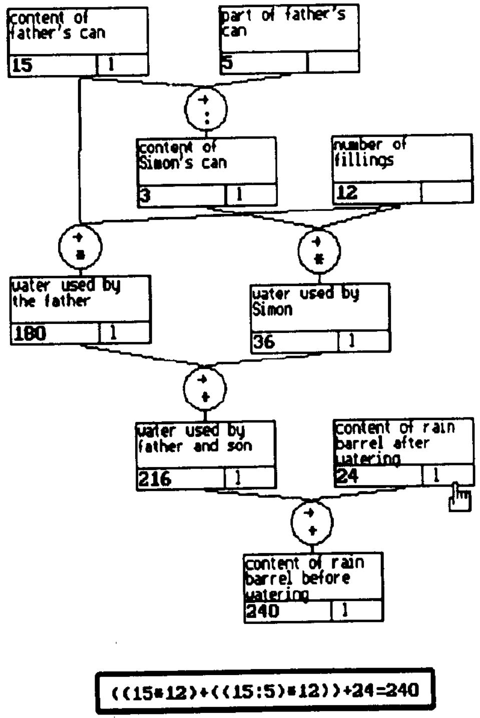


FIG. 5.6a. Two different solution trees mirroring two paths of comprehension of the same task: Little Simon and his father are watering their vegetable garden. The father has a 15-liter watering can. Simon's can holds one fifth of that. Both fill their cans 12 times. After that, there are still 24 liters in the rain barrel. How much water can the rain barrel hold? (translation from German)

- they illuminate the hidden construction processes by which the student determines the structure of problem situations;
- in so doing they make students' thinking overt and accessible to reflection and discussion;
- they encourage generative understanding, i.e., one can start constructing a tree without already having completely understood the problem;
- they provide an efficient form of written calculus that can be directly translated into informationally equivalent equations;
- they provide a bridging representation intermediate to both text and implicit mathematical structure;

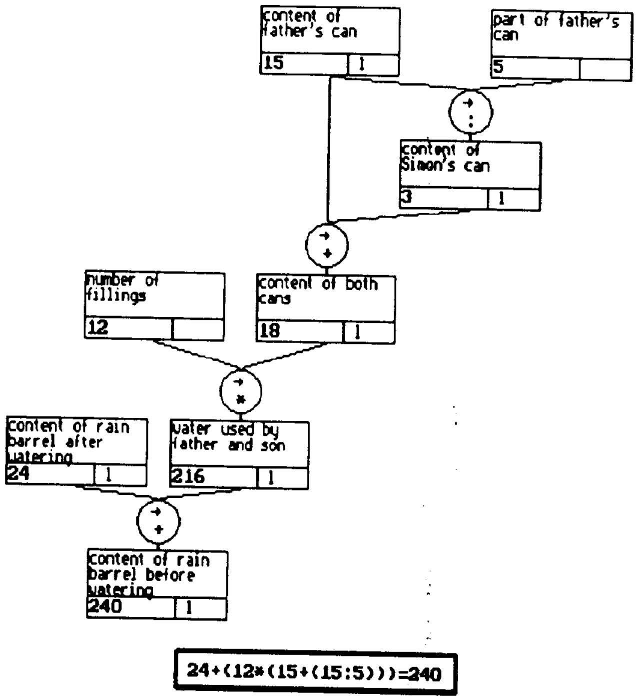


FIG. 5.6b. *(Cont.)*

- they facilitate movement between textual (situational) and mathematical understanding;
- they are usable for students from a very early age;
- they provide a basis for the design of feedback.

The concept of solution or calculation trees was theoretically derived from an earlier notion of *Simplex-Complex structures* (Breidenbach, 1963; Fricke, 1987), introduced as *thinking tools* (Bauersfeld, 1965) into mathematics education in the '60s. Bauersfeld defined a *simplex* as "a relational network across three mathematically relevant domain-concepts" (p. 125), whereas a *complex* is a combination of simplex structures. Solution trees are used in Swiss and German mathematics text books as an intelligible, but hardly flexible tool (with respect to pencil and paper manipulation).[9]

[9]A formalism similar to the framework of solution trees was proposed by Shalin and Bee (Greeno, 1987), and further discussed by Hall, Kibler, Wenger, and Truxaw (1989). In Shalin and

Solution trees allow simultaneous grasping of both the semantic (situational) and the mathematical deep structure denoted by a problem text. By following the different converging semantic and mathematical constraints propagated through the network structure of a solution tree, the corresponding problem can be understood on three levels: on a purely *semantic level of domain relations* (Sachverhältnisse); on the level of *numbers and quantitative operations* that progressively constrain the mathematical unknown; and finally, on the level of *units of measurement.*

With regard to instruction, the crucial aspect of solution trees is that they do not allow one to focus only on the latent or implicit mathematical structure of a problem; that is, they are tools of both quantitative and qualitative reasoning. By providing subfields of information describing the semantic role of every quantitative entity processed in a tree and relating these to the problem as a whole, students are forced to reconstruct a problem at its mathematical *and* semantic (linguistic-situational) level. In HERON the labels that interpret the semantic meaning of a quantity in the qualitative context of an episodic situation, or action, are called *situation concepts.* To find adequate and concise situation concepts, which give episodic meaning to any quantity employed in the construction of a tree, is not a trivial task. Good situation concepts are both products of task analysis and conceptual tools for the synthesis of the solution. While the network of quantities in a tree provides a mathematical interpretation, the network of situation concepts provides a semantic interpretation of the problem.

Solution trees and the tool kit for their construction in HERON not only give students a way to express graphically what they *think* the content-specific mathematical deep structure of a wide class of problems is, but also provide students with a constraining *schematic format* or a control structure for how the mathematical problem model ought to look. That is, by means of their schematic properties, solution trees provide a *structural form* for the planning and construction of mental models. While the mathematical understanding of a problem situation takes shape in a student's head, the schematic form of the solution tree serves as a perceptual constraint that must be satisfied by overtly constructing it on the screen of the computer.

Bee's conceptual language (which inspired also work by Thompson, 1990), networks of quantitative entities are used to describe the quantitative forms of classes of mathematical story algebra problems. Differences with the representational format used in HERON have to do with how mathematical entities and operations are treated. To my understanding, Shalin and Bees' formalism is closer to a more static and formalist view of mathematical structure, whereas solution trees, with their explicit notation of mathematical operations, are closer to an operative view of mathematical thinking (Aebli, 1980; Piaget, 1970). According to Piaget, mathematical operations are the developmental derivatives of certain classes of sensorimotor actions bearing an abstract mathematical meaning to be expressed by the set of elementary mathematical operations. However, it is beyond the scope of this chapter to provide a detailed comparison of the two conceptual approaches.

A Sample Session with HERON

HERON has been designed to facilitate the solving of any story problem that can be represented by solution trees. There are two implementations of the system: A prototype version is written in Loops and runs on a Xerox workstation (Kämpfer, 1991); another version designed for use in classrooms is written in C and runs on IBM-type machines (Stüssi, 1991).

Only a little instruction is needed to achieve almost full use of the functions in HERON. It takes third graders about 20 minutes to become familiar with the entirely mouse-driven interface of the system. Most commands and graphical tools are available as buttons displayed on the screen. Some important menus, as, for example, for filling situation concepts in solution trees, can be activated by pushing a mouse button in an appropriate, active region of the screen. In order to demonstrate the functioning of the system, the following section describes two examples of how students can use it.

The problem text of the first example is shown in Fig. 5.6. Previous to any overt construction, the student selects a problem and reads it. Following reading, the student is asked by the computer to identify relevant quantitative information in the problem text. This is done by highlighting numbers or number-placeholders in the text with the mouse-cursor. When a piece of numerical information is selected, the system creates a graphical situation unit with the selected number already filled in. The student is then asked to fill in a unit of measurement (for example *liters*) and a textual label (for example *content of father's can*). The label can be selected as a whole from a menu, or it can be constructed from a list of word elements from a menu. After the situation unit has been completed, a new piece of information is selected from the text. Depending on how the system is initialized (e.g., for weak or novice students), the student gets an error message if he or she selects a piece of irrelevant information from the text or fails to select a relevant piece. In the standard (non-novice) condition, the student decides on his or her own when to stop selecting numbers and creating situation units.

After setting up some situation elements, the student can start planning and constructing a tree-structure for the solution (Fig. 5.7). In order to create and instantiate a preliminary relational schema, for example, in order to achieve the subgoal of computing the content of Simon's can, the units labeled *content of father's can* and *part of father's can* are selected with the mouse, moved to the upper left corner of the screen and linked together. The latter is done by selecting, placing, and linking an empty operator node (circle) and two line segments. With the selection of an operator node, an empty box is automatically generated by the system. That is, for every pair of situation elements linked by an empty operator slot (circle), the system completes the nascent relational schema by creating an empty element or subgoal-slot. Before the user, with the help of menus, can fill the unit of measurement and the label into the emerging third element, he/she is asked to select the appropriate mathematical operation from a

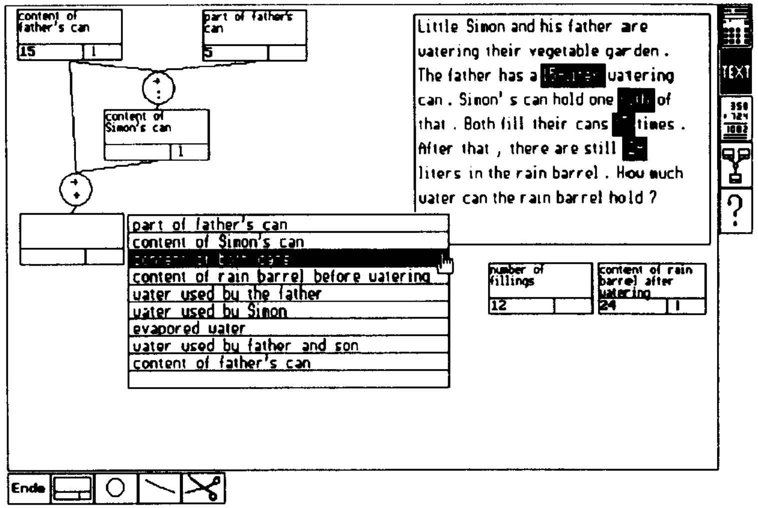


FIG. 5.7. Partial solution tree of the Simon task (for problem text see Fig. 6; implementation on IBM AT compatible; translation from German).

menu which appears by pushing the mouse on the operator node. Now the student can fill in the label (*content of Simon's can*) and the unit of measurement (*liters*) and decide if he/she wants to calculate the subgoal immediately. The calculation can be done with or without system support: for example, clicking on the numerical field lets the system calculate and display the correct result in the numerical field.

After each relational schema is instantiated, its resulting element can be used to generate new schemata and to achieve further subgoals. It is up to the student to choose which comprehension path to follow, that is, how to navigate through the problem space. In order to determine the intermediate level subgoal of the *total amount of water carried by father and son*, there are two main paths open. In the first path (Figs. 5.7 and 5.8), an additive schema is first generated which computes the content *of both watering cans*, and then a multiplicative schema is instantiated computing the *total amount of water*. In a second, slightly more complicated path (depicted in Fig. 5.6a), the order of mathematical operations is reversed: The instantiation of two multiplicative schemata

MULTIPLY (number of fillings, content of father's can)
MULTIPLY (number of fillings, content of Simon's can)

is followed by an additive schema

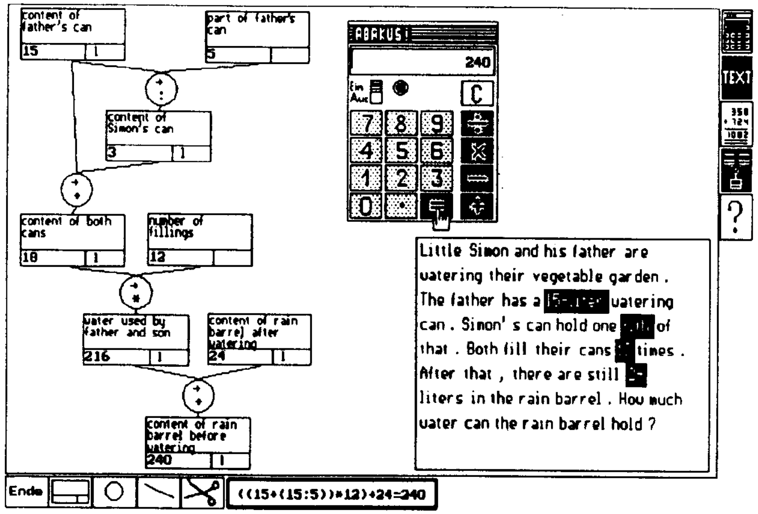


FIG. 5.8. Complete solution tree of the Simon task.

ADD (amount of water carried by the father, amount of water carried by the son)

leading to the same intermediate result (216 liters). Relating the *total amount of water carried* to the *amount of water remaining in the rain barrel* (24 liters) and computing the final schema leads to the final result of the *amount of water in the rain barrel before watering* (240 liters).

In order to generate equations from the solution tree, the student can click the right side of the mouse-button on any numerical value in the tree. The system then displays the equation with the selected value. For example, clicking on the value "216" in Fig. 5.6a would produce the partial equation

$$(15 * 12) + (15:5) * 12 = 216,$$

while clicking on the same value in Fig. 5.6b would produce the following equation:

$$12 * (15 + (15:5)) = 216.$$

And clicking on the final value of "240" in Fig. 5.8 will produce the goal-generating equation

$$((15 + (15:5)) * 12) + 24 = 240.$$

An example of a much more complex problem is the *Afghanistan* task. It was developed by Aebli and Staub (1985) as a new type of authentic and rich situation problems to be used in mathematics education. The elaborated story contains a nontrivial episodic, that is nonmathematical structure, describing an agent who is planning and executing a complex action leading from an initial state to an end state. The successful execution of each of its parts requires that certain situational conditions, which are either produced or encountered by the agent, be fullfilled.

Afghanistan is a mountainous country, just like Switzerland, but much drier. There are fewer springs and streams, and water is rare and valuable. From a small village a muletrack leads over a pass into a small town. To drive animals into town, one needs several days. At the end of the first day, one comes to an alp. Here a shepherd boy keeps 18 sheep and 15 goats. Above the alp, there is a snowfield which melts in the course of the summer. The water then flows towards the alp and gives 110 liters of drinking water a day for the animals. But now, the snow has not melted. Until it does, the animals drink from a waterhole below the rocks in which the water from a spring is collected. The hole is not leak-proof, so 35 liters of water ooze away every day. Each sheep drinks about 5.4 liters a day, each goat 3.8 liters.

The shepherd drinks about 2.3 liters of goatmilk a day. The spring yields 350 liters of water a day. Every day 5 liters evaporate. Each evening the shepherd drains the waterhole. He leads all the water that is left through a small canal into a basin. He has just done that this evening. He estimates that there are 90 liters of water in the basin at the moment. Another 120 liters of water ooze away as it flows to the basin. It is not lost, however. It irrigates the meadow below the small canal. As a result, more grass grows here. The shepherd can cut it 3 times a year and obtains 150 kilograms of hay each time. The hay is brought to the village and fed to the animals during wintertime. The basin is fenced with poles, so the sheep and the goats cannot drink from it. From time to time, the peasants drive some cows from the village over the pass into the town and sell them at the market. On the first day, they find water on only at the alp. When they get there, the shepherd boy takes the poles away from the basin and lets the cows drink from it. On the day on which he has estimated the available water, his father arrives at the alp late in the evening. He says to his son: "I want to bring 24 cows to the market as soon as possible. I think that they'll need a total of 320 liters of water up here." The boy says: "I can calculate when you can come." When can his father come with the cows?

As with the mathematization of any demanding word problem, the solving of the Afghanistan task requires the implicit quantitative structure to become explicit as an interconnected system of mathematical relations. However, before the underlying quantitative structure can be generated in its ultimate representational format of a solution equation, one has to understand, in a constructive effort after meaning, the problem text, the situation denoted by the text, and the implicit mathematical relations involved. The complexity of these processes of understanding, and the fact, that they require sequential and hierarchical planning,

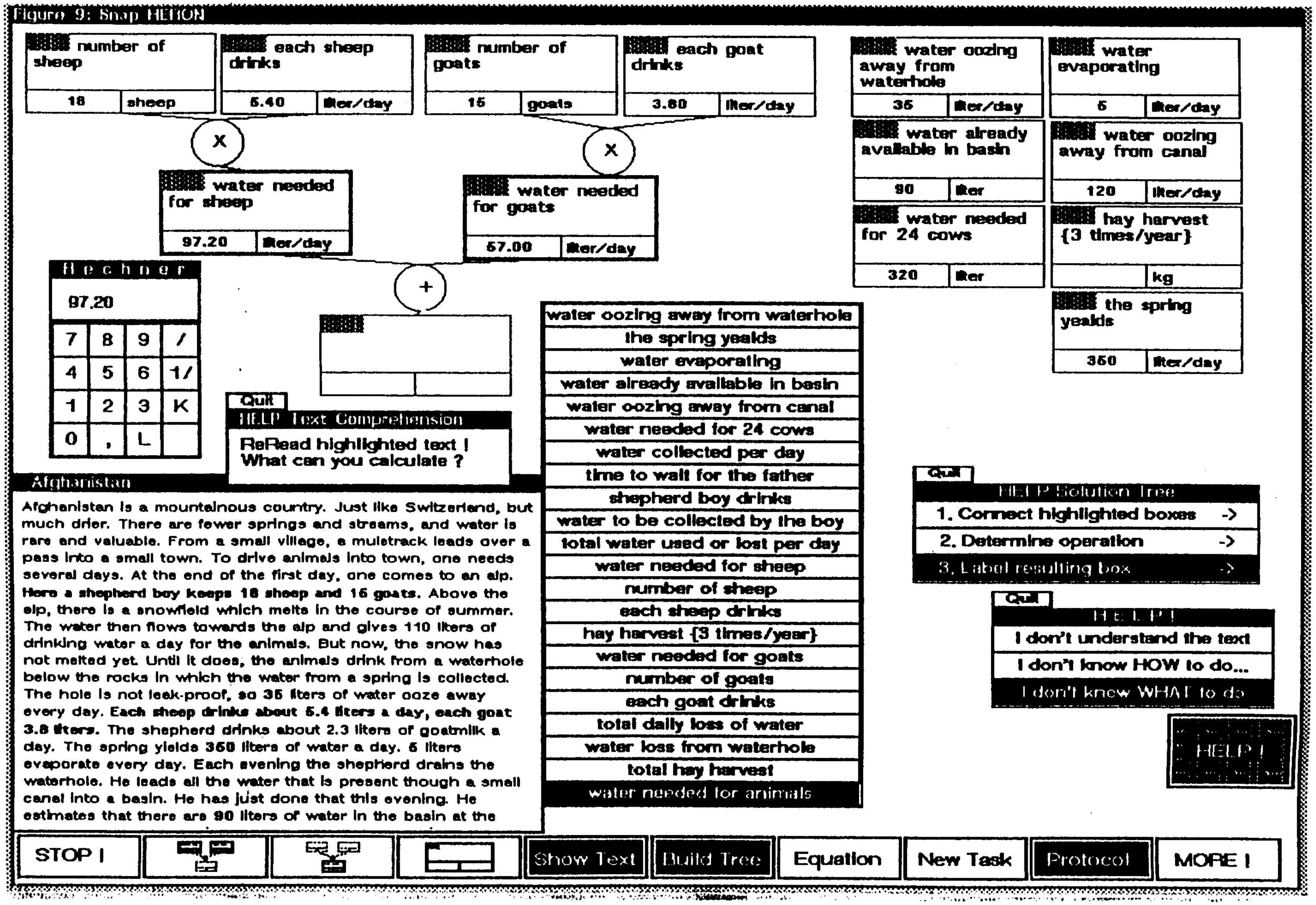


FIG. 5.9. Afghanistan task: Partial solution tree (Implementation on Xerox workstation 1186; translation from German)

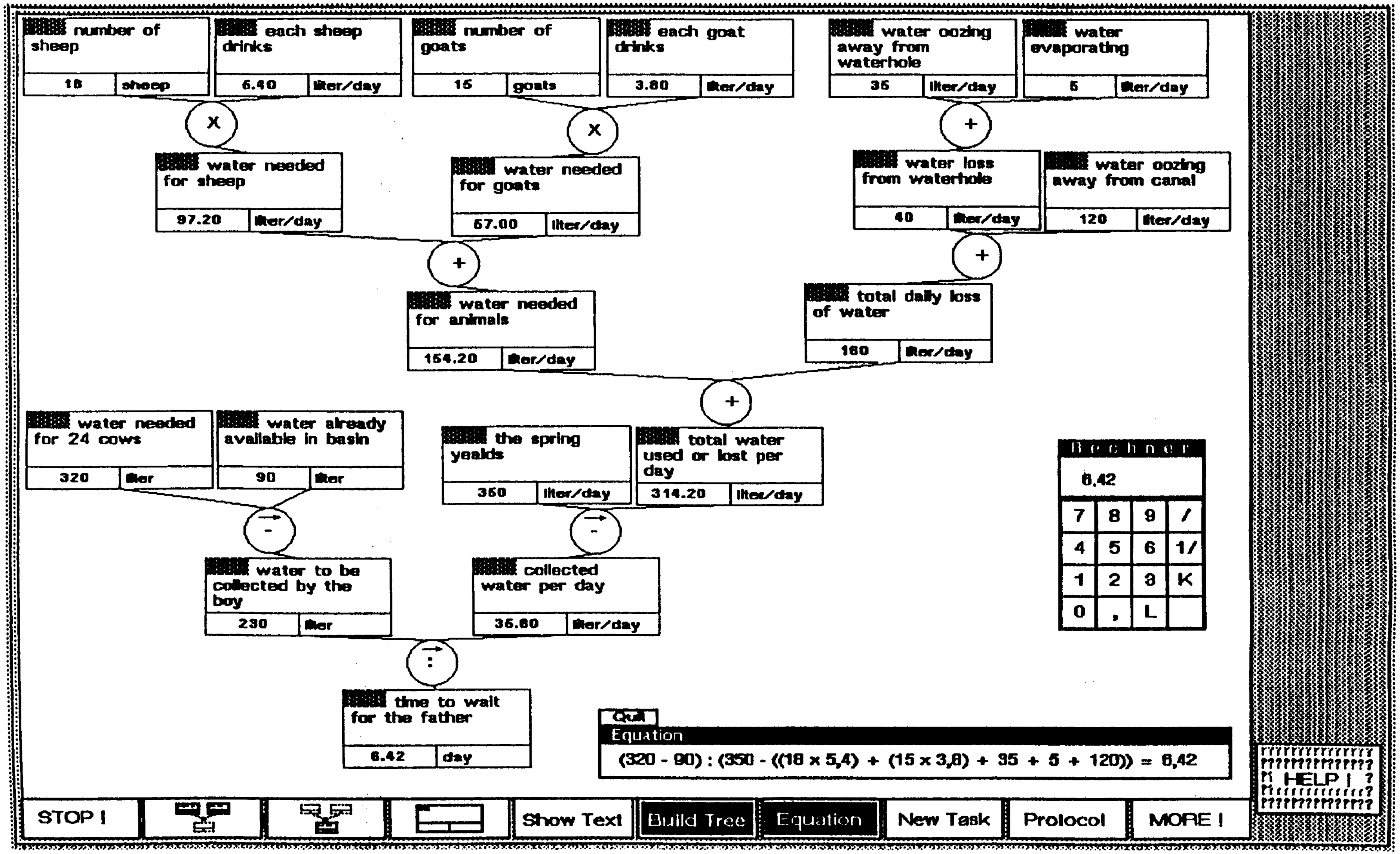


FIG. 5.10. Afghanistan task: Completed solution tree.

makes solution or planning trees especially suitable to the solving of semantically rich action problems like the Afghanistan task.

Figure 5.9 shows a solution tree (one among several possible trees) *in statu nascendi*, while Fig. 5.10 shows the same tree in its completed form as a coherent and hierarchical system of mathematical relations to be mapped onto a corresponding solution equation. From the operational perspective set up by the quantitative problem question, this solution equation represents the most concise synoptic form of the relevant mathematical aspects of the complex action.

While using HERON the student can call on a *HELP* component for both deeper comprehension of the problem text and for the construction of solution trees. *Text comprehension aids* mainly consist of explanations of words, paraphrasing of sentences, and of giving hints about which text fragments belong together and should be processed together. *Construction aids for solution trees* operate in conjunction with the aids for text comprehension, and provide help in operating the system (how to use the tools of the interface), with planning, and with constructing the trees (cf. Kämpfer, 1991; Stüssi, 1991). With the latter type of help the system provides the student (on request) with progressively more detailed hints about what to do next, ultimately offering a next operative step.

With respect to *feedback*, HERON is unintelligent, but nevertheless fairly powerful. The system does not perform cognitive diagnosis based on individualized student modeling. So far, it knows nothing about the thinking of individual students while solving a problem. However, HERON does know how to build conceptual networks, and what students are *overtly doing* when constructing their solution trees. HERON is thus able to provide feedback and customized help to the individual student on the basis of a detailed task analysis. With respect to tasks, HERON knows which quantified situation elements can or must (not) be connected in a solution tree, and which labels and units of measurement are to be attached to each quantified situation unit. HERON is able to monitor the students' overt constructive activity of planning and creating a solution tree, and to provide feedback on four types of errors: (a) mathematical operation errors (two situation units are connected by a false operation), (b) labeling errors, if an incorrect situational role concept is attached to a quantity box or situation unit, (c) errors on the selection of units of measurement, and (d) errors regarding the false inclusion or ommission of (ir)relevant mathematical information in a solution tree.[10]

Ongoing Work: The Use of HERON for Instruction

The Afghanistan task gives an impression of the constructive power and flexibility of HERON, which allows the student to decompose a problem into relational

[10]A different and more "intelligent" approach to error handling is used by Derry and Hawkes (1989) in the TAPS system, which is very similar to HERON in its representational format. TAPS treats errors as deviations from recognizable and preset ideal solution paths.

building blocks or subgoals and to create solution trees of almost any complexity and in any arbitrary sequence in which subgoals can be achieved. The task also illustrates the diverse and demanding *language and situation comprehension activities* that are required in solving mathematical word problems (Staub & Reusser, 1991). HERON thus can be applied as a cognitive tool to the two fundamental curricular tasks of schooling, mathematics education and reading or language comprehension instruction. Moreover, HERON can also be considered as a preparatory tool for algebra instruction: The combination of the hierarchical nature of solution trees, which are easily translatable into equations, and the system's ability to form a parameter for any given number in a problem, makes it possible to redescribe any solution tree in terms of the assigned parameters.

For the most part, the students in HERON work in pairs. After an instructional phase of 10 to 20 minutes that includes the construction of a sample solution tree, students are able to construct the solution tree for a simple problem. Complex tree, such as the Afghanistan task, may take much longer. In constructing solution trees, most students thus far have followed a *modus operandi of forward chaining* similar to the strategy described by Derry and Hawkes (1989): After a first relational unit is schematically constructed (by drawing three boxes or situation units), the boxes are numerically and situationally interpreted by filling in the numbers (one of them to be computed), the units of measurement, and the situation concepts. Then, the next schema is constructed that includes the previous result set as one of its elements. The construction—proceeding stepwise from the given information to the anticipated goal state—continues in this manner until the tree is completed.

HERON records in a log file all activities performed on the screen, including time characteristics and turn taking with the mouse. This allows *the comprehension (solution) paths to be replayed*. Pairs of students can reflect upon and discuss their solution paths. Replays are used to elicit more complete retrospective reports on what the students were thinking while solving the problem.

HERON is currently being evaluated in classroom settings that encourage collaborative problem solving in groups or pairs. It is also being tested in an empirical study comparing the solving of word problems by single users and by pairs *with and without HERON*. We think that pairs of students working together can support each other in many ways beneficial to each other. On the one hand, pairs of students share the diverse activities directly related to solving the problems, such as jointly constructing understanding of problems, monitoring and refining each other's constructions of mental models, or discussing alternative solution paths. By working together they are also able to use the system more effectively. Computer-based learning environments may still be relatively weak, compared to a sensitive human teacher, and thus require multiple and combined expertise—from the system's own (modest) intelligence and from intelligent users alike.

ACKNOWLEDGMENTS

The work in this paper was supported by the Swiss National Science Foundation (Grant No. 10-2052.86. I wish to thank the members of our research group, in which many of the ideas in this paper were shaped and discussed: Xander Kämpfer, Markus Sprenger, Fritz C. Staub, Rita Stebler, and Ruedi Stüssi. I equally appreciate the many invaluable comments on the writing of this chapter from Eileen Kintsch, as well as those of the Editors. Ruedi Stüssi's and Xander Kämpfer's assistance in preparing the figures of the chapter is also gratefully acknowledged.

REFERENCES

Aebli, H. (1961). *Grundformen des Lehrens*. Stuttgart: Klett.

Aebli, H. (1978). Von Piagets Entwicklungspsychologie zur Theorie der kognitiven Sozialisation. In G. Steiner (Ed.), *Piaget und die Folgen*. Zürich: Kindler.

Aebli, H. (1980, 1981). *Denken. Das Ordnen des Tuns* (Two vols.). Stuttgart: Klett.

Aebli, H. (1983). *Zwölf Grundformen des Lehrens*. Stuttgart: Klett.

Aebli, H., & Staub, F. C. (1985, August). *From "text comprehension" to the mathematical comprehension of text*. Paper presented at the First Convention of the European Association for Research on Learning and Instruction EARLI, Leuven, Belgium.

Aebli, H., Ruthemann, U., & Staub, F. C. (1986). Sind Regeln des Problemlösens lehrbar? *Zeitschrift für Pädagogik, 32*, 617–638.

Anderson, J. A., Boyle, C. F., Farrell, R., & Reiser, R. (1984). Cognitive principles in the design of computer tutors. *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*, Boulder, Colorado.

Barker, R. C. (1978). Theory of behavior settings. In R. C. Barker (Ed.), *Habitats, environments, and human behavior*. San Francisco: Jossey-Bass.

Bartlett, F. C. (1932). *Remembering*. London: Cambridge University Press.

Bauersfeld, H. (1965). Der Simplex-Begriff im Sachrechnen der Volksschule. *Die Schulwarte, 18*, 124–132.

Beck, E. (1989). Eigenständiges Lernen. *Beiträge zur Lehrerbildung, 7*, 169–178.

Bobrow, D. G. (1964). *Natural language input for a computer problem solving system*. Unpublished doctoral thesis, MIT.

Breidenbach, W. (1963). *Rechnen in der Volksschule*. Hannover: Schroedel Verlag.

Brown, A. L., & Palincsar, A. S. (1989). Guided, cooperative learning and individual knowledge acquisition. In L. B. Resnick (Ed.), *Knowing, learning and instruction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Brown, A. L., & Campione, J. C. (1990). Interactive learning environments and the teaching of science and mathematics. In M. Gardner, J. G. Greeno, F. Reif, A. H. Schoenfeld, A. DiSessa, & E. Stage (Eds.), *Toward a scientific practice of science education*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Bruner, J. (1961). The act of discovery. *Harvard Educational Review, 31*, 21–32.

Bruner, J. (1986). *Actual minds, possible worlds*. Cambridge, MA: Harvard University Press.

Bruner, J. S. (1990). *Acts of meaning*. Cambridge, MA: Harvard University Press.

Bruner, J. S., Olver, R. R., & Greenfield, P. M. (1966). *Studies in cognitive growth*. New York: Wiley.

Clancey, W. (1989). Situated cognition and intelligent tutoring systems. In H. Maurer (Ed.), *Computer-assisted learning*. New York: Springer.

Collins, A., & Brown, J. S. (1988). The computer as a tool for learning through reflection. In H. Mandl & A. Lesgold (Eds.), *Learning issues for intelligent tutoring systems*. New York: Springer.

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, learning and instruction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Cumming, G., & Self, J. (1990). Intelligent educational systems: identifying and decoupling the conversational levels. *Instructional Science, 19*, 11–27.

Cummins, D., Kintsch, W., Reusser, K., & Weimer, R. (1988). The role of understanding in solving word problems. *Cognitive Psychology, 20*, 439–462.

De Corte, E., Verschaffel, L., & de Win, L. (1985). Influence of rewording problems on Children's problem representations and solutions. *Journal of Educational Psychology, 77*, 460–470.

Derry, S. J., & Hawkes, L. W. (1989). *Error-driven cognitive apprenticeship: A feasible ITS approach*. Florida State University, Tallahassee.

Duncker, K. (1935). *Zur Psychologie des produktiven Denkens*. Berlin: Springer.

Fricke, A. (1987). *Sachrechnen, Das Lösen angewandter Aufgaben*. Stuttgart: Klett.

Glaser, R. (1987). The study of cognition and instructional design: Mutual nurturance. *Behavioral and Brain Sciences, 10*, 483–84.

Greeno, J. G. (1983). Conceptual entities. In D. Gentner & A. L. Stevens (Eds.), *Mental models*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Greeno, J. G. (1987). Instructional representations based on research about understanding. In A. H. Schoenfeld (Ed.), *Cognitive Science and mathematics education*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Hall, R., Kibler, D., Wenger, E., & Truxaw, C. (1989). Exploring the episodic structure of algebra story problem solving. *Cognition and Instruction, 6*, 223–283.

Hollenstein, A., Staub, F. C., & Stüssi, R. (1987). Was passiert, wenn . . . Computersimulation als didaktisches Hilfsmittel. *Schweizer Schule, 10*, 10–16.

Hudson, T. (1983). Correspondences and numerical differences between disjoint sets. *Child Development, 54*, 84–80.

Hutchins, E. L., Hollan, J. D., & Norman, D. A. (1985). Direct manipulation interfaces. In D. A. Norman & S. W. Draper (Eds.), *User centered system design: New perspectives on human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Kämpfer, A. (1991). *Prototypische Entwicklung eines adaptiven tutoriellen Systems zum Verstehen und Lösen mathematischer Textaufgeben*. Lizentiatsarbeit. University of Bern, Switzerland.

Kaput, J. J. (1989). Linking representations in the symbol systems of algebra. In S. Wagner & C. Kieran (Eds.), *Research issues in the learning and teaching of algebra*. Reston, VA: The National Council of Teachers of Mathematics.

Kintsch, W. (1974). *The representation of meaning in memory*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Kintsch, W., & Greeno, J. G. (1985). Understanding and solving word arithmetic problems. *Psychological Review, 92*, 109–129.

Klafki, W. (1963). *Das pädegogische Problem des Elementaren und die Theorie der kategorialen Bildung*. Weinheim: Beltz.

Larkin, J. M., & Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science, 11*, 65–99.

Lehtinen, E. (1989). Verstehen lehren als Verändern von Lern- und Bewältigungsstrategien. *Beiträge zur Lehrerbildung, 7*, 213–219.

McArthur, D., Stasz, C., & Zmuidzinas, M. (1990). Tutoring techniques in algebra. *Cognition and Instruction, 7*, 161–195.

Mead, G. H. (1934). *Mind, self, and society*. Chicago: Chicago University Press.

Nathan, J. N., Kintsch, W., Young, E. (1990). *A theory of algebra word problem comprehension and its implications for unintelligent tutoring systems* (Technical Report 90-02). Institute of Cognitive Science, University of Colorado, Boulder.

Nesher, P. (1989). Microworlds in mathematical education: a pedagogical realism. In L. B. Resnick (Ed.), *Knowing, learning and instruction*. Hillsdale, NJ: Lawrence Erlbaum Associates.
Ohlsson, S. (1986). Some principles of intelligent tutoring. *Instructional Science, 14*, 293–326.
Ohlsson, S. (1987). Sense of reference in the design of interactive illustrations for rational numbers. In R. W. Lawler & M. Yazdani (Eds.), *Artificial intelligence and Education*. Norwood, NJ: Ablex.
Paige, J. M., & Simon, H. A. (1966). Cognitive processes in solving algebra word problems. In B. Kleinmuntz (Ed.), *Problem solving*. New York: Wiley.
Pea, R. D. (1987). Cognitive technologies for mathematics education. In A. H. Schoenfeld (Ed.), *Cognitive Science and mathematics education*. Hillsdale, NJ: Lawrence Erlbaum Associates.
Piaget, J. (1947). *La psychologie de l'intelligence*. Colin: Paris.
Piaget, J. (1970). *Genetic epistemology*. New York: Columbia University Press.
Piaget, J. (1973). *To understand is to invent*. New York: Grossman.
Polanyi, M. (1966). *The tacit dimension*. London: Routledge & Kegan.
Resnick, L. B., & Johnson, A. (1988). *Intelligent machines for intelligent people: Cognitive theory and the future of computer-assisted learning*. Pittsburgh: Learning Research and Development Center.
Reusser, K. (1984). *Problemlösen in wissenstheoretischer Sicht*. Unpublished doctoral dissertation. University of Bern, Switzerland.
Reusser, K. (1985). *From text to situation to equation* (Technical Report). Institute of Cognitive Science. University of Colorado, Boulder.
Reusser, K. (1988). Problem-solving beyond the logic of things: Contextual effects on understanding and solving word problems. *Instructional Science, 17*, 309–339.
Reusser, K. (1989a). *Vom Text zur Situation zur Gleichung. Kognitive Simulation von Sprachverständnis und Mathematisierung beim Lösen von mathematischen Textaufgaben*. Habilitation Thesis. University of Bern, Switzerland.
Reusser, K. (1989b, September). *Textual and situational factors in mathematical word problems*. Paper presented at the Third Conference of the European Association for Research on Learning and Instruction. Madrid/Spain.
Reusser, K. (1990a). From text to situation to equation: Cognitive simulation of understanding and solving mathematical word problems. In H. Mandl, E. De Corte, N. Bennett, & H. F. Friedrich (Eds.), *Learning and instruction in an international context* (Vol. II). New York: Pergamon Press.
Reusser, K. (1991, August). *Intelligent technologies and pedagogical theory: Computers as tools for thoughtful teaching and learning*. Invited address at the Fourth Conference of the European Association for Research on Learning and Instruction (EARLI). Turku, Finland.
Roschelle, J. (1990, April). *Designing for conversations*. Paper presented at the meeting of the American Educational Research Association, Boston.
Salomon, G. (1990, August). *The changing role of the teacher: From knowledge transmitter to exploration orchestrator*. Paper presented at the International Symposium on Research on Effective teaching and responsible learning. Fribourg, Switzerland.
Salomon, G., Perkins, D. N., & Globerson, T. (1991). Partners in cognition: Extending human intelligence with intelligent technologies. *Educational Research, 20*(3), 2–9.
Scardamalia, M., Bereiter, C., McLean, R. S., Swallow, J., & Woodruff, E. (1989). Computer-supported intentional learning environments. *Journal of Educational Computing Research, 5*, 51–68.
Schoenfeld, A. (1989). Problem solving in context(s). In R. I. Charles & E. A. Silver (Eds.), *The teaching and assessing of mathematical problem solving*. Reston, VA: National Council of Teachers of Mathematics.
Selz, O. (1922). *Zur Psychologie des produktiven Denkens und des Irrtums*. Bonn: Cohen.
Simon, H. A. (1977). *Models of discovery and other topics in the methods of science*. Dordrecht: Reidel.
Sleeman, D., & Brown, J. S. (1982). *Intelligent tutoring systems*. New York: Academic Press.

Staub, F. C., & Reusser, K. (1991, August). *What makes mathematical word problems difficult? The role of presentational factors*. Paper presented at the Fourth Conference of the European Association for Research on Learning and Instruction (EARLI). Turku, Finland.
Stüssi, R. (1991). *WATGRAF. Ein tutorielles System zum Lösen von handlungsbezogenen Sachaufgaben*. Lizentiatsarbeit. University of Bern, Switzerland.
Thompson, P. W. (1990). *Word Problem Analyst. Version 2.2. Tutorial I & II*. Center for Research in Mathematics and Science Education. San Diego State University.
Van Dijk, T., & Kintsch, W. (1983). *Strategies of discourse comprehension*. New York: Academic Press.
Van Hiele, P. M. (1986). *Structure and insight. A theory of mathematics education*. Orlando, FL: Academic Press.
Van Lehn, K. (1988). Student modeling. In M. Polson & J. J. Richardson (Eds.) *Foundations of intelligent tutoring systems*. Hillsdale, NJ: Lawrence Erlbaum Associates.
Verschaffel, L. (in press). A decade of research on word-problem solving in Leuven: Theoretical, methodological, and practical outcomes. *Educational Psychology Review*.
Vygotsky, L. S. (1978). *Minds in society: The development of higher psychological process*. Cambridge, MA: Harvard University Press.
Wenger, E. (1987). *Artificial intelligence and tutoring systems*. Los Altos, CA: Kaufmann.
Wertheimer, M. (1945). *Productive thinking*. New York: Harper.
Winston, P. H. (1984). *Artificial intelligence*. Reading, MA: Addison-Wesley.