International Journal of Technology and Design Education **8**, 167–184, 1998.

Framework for the Study of Cognitive and Curricular Issues of Technological Problem Solving

D. MIODUSER

Tel-Aviv University, School of Education, Ramat-Aviv 69978, Israel

ABSTRACT: Technology education has undergone extraordinary changes in the last decade. Educators and educational policy makers have become aware of the importance of *technology* in the basic formation of today's educated person. At the same time rapid and continuous change in technology itself poses serious challenges in regards education and training in this field. Thus technology related contents, skills and teaching strategies for all specialization levels are being reconsidered and redefined for educating both the technologically literate person and the expert practitioner. In an attempt to examine the learning process of technological problem solving (TPS), we suggest and define an appropriate conceptual framework, encompassing all components of the process, i.e., knowledge, skills, and cognitive models. First the background to our proposal is presented, followed by the description of the two main components of the model: the Learning Space and the Technological Primitives repertoire.

Keywords: cognitive models, design process, technological literacy, technological primitives, technological-problem-solving

Imagine a class discussion taking place as part of the technology curriculum learning activities. The students are presented with a picture showing a series of objects created by the Japanese artist Shigeo Fukuda (Figure 1), including a three-blades scissors, a branching screw, and other 'impossible objects'.

The very first issues for discussion could be: 'What are these objects?'; 'What are they for?'; 'Do we know their names, or can we suggest appropriate names?'. Next we might be asking about their functioning: 'How do we use them?'; 'How do they work?'. Very rapidly we may arrive at wondering: 'Do they work at all?'; 'Why not?'; 'What's wrong?'. Once we have diagnosed what's wrong with these objects we may next ask: 'How do we get them to work?'; 'What changes should be made?'. And on a yet further level of sophistication we may ask: 'Are there ways to use these objects as they are (e.g., by inventing a player for triangular records)?'; or 'What other solutions (for performing similar functions) can be designed to replace these impossible objects?'.

The above questions suggest a hierarchical progression in the analysis of the objects. They imply the activation of different cognitive processes (e.g., a description of the object as opposed to the description of a fault in it). They imply different knowledge levels (e.g., intuitive sense that it will not work vs. a well-founded explanation of why it does not work, or suggestions for replacement of inappropriate components in the design). The above questions also imply the activation of different skill levels (e.g., looking for *relevant* features while classifying an object within artifactual

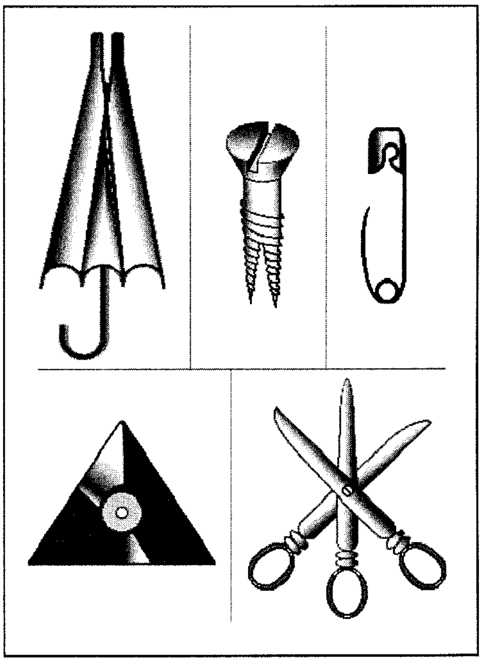

Figure 1. Collection of 'Impossible objects' by Shigeo Fukuda.

categories, or a search for *desired* features in candidate replacement parts for faulty components). Moreover they relie on different internal (cognitive) representations of the object under consideration (e.g., structural features for recognition purposes as opposed to a causal/functional model for predicting the object's functioning in various conditions).

To examine the learning process of technological problem solving (TPS), we suggest the definition of an appropriate conceptual framework, encompassing all components of the process, i.e., knowledge, skills, and cognitive models. This paper presents such a framework for the characterization of Technological Problem Solving. First we present the background to our model and subsequently we describe its two main components: the Learning Space and the Technological Primitives repertoire.

BACKGROUND

The nature and desired features of the technological-problem-solver's knowledge and skills have been studied in a wide range of published works. These include theoretical considerations (e.g. Gropius, 1975; Simon, 1981; Ihde, 1990, 1993; Chen & Strupp, 1993; Mitcham, 1994), research results (e.g., Wu et al., 1996; McCormick et al., 1994; Mioduser et al., 1996) teaching-practice oriented publications (e.g., Savage & Sterry, 1990), and curricular reports and materials (e.g., Nuffield, 1995; Mioduser, 1998). We will focus here on several key ideas and questions relevant to the model to be proposed.

At the epistemological level, Technology is generally presented as a discipline in its own right, as a body of knowledge with its unique history and development, philosophy, contents, and methodology (e.g., Ihde, 1993; Stevens, 1995; Chen & Strupp, 1993). The nature of its interrelations with other areas of human knowledge has also been studied (e.g., with science in Gardner, 1995; de Vries, 1996; or with art in Vos, 1996). These epistemological elaborations are crucial in that they address the main issues of any curricular enterprise or research into learning in technology education: What are the contents of technology education, and what are the links between these contents and other knowledge areas? How can the boundaries of 'technological literacy' and 'technological expertise' knowledge be defined in epistemological as well as in instructional terms?

In terms of skills and methodology published work focuses on sets of skills and processes which characterize the thinking and practice of technology. Some work characterizes technological problem solving (TPS) processes by professional practitioners (e.g., Vincenti, 1990; Bucciarelli, 1996). Other focuses on the overall set of skills expected to be part of the cultural background of any technologically literate person (e.g., Dyrenfurth, 1991). A claim which enjoys widespread consensus is that the key methodological construct of the field is the 'design process', the systematic process by which TPS takes place (e.g., Bucciarelli, 1996, pp. 111–116). These considerations at the methodological level serve as background for the discussion of an additional set of questions regarding technology education: What groups of skills and processes are relevant for what populations? And for different kinds of problems? How malleable is the generic design-process model as to allow meeting the needs of different populations and types of problems?

As concerns cognitive and learning processes attempts have been made to create models in support of research and instruction which reflect the cognitive processes involved in solving technological problems (e.g., Levin & Mioduser, 1996). Empirical work focuses on learning processes at different levels, e.g., school children learning technology (e.g., McCormick et al., 1994), or technicians acquiring specific skills (Lesgold et al., 1992). Representative issues dealt with in this area are the following: What conceptions, misconceptions and missing conceptions of technology do learners (with different cognitive levels and needs) hold? How do these affect the acquisition and application of knowledge and skills? What are the similarities and differences between intuitive and systematically acquired problem solving skills (e.g., see Wu et al., 1996)?

At the instructional level, published work deals with different issues, such as the evolution of technology education (e.g., Stevens, 1995; Grubb, 1996), models for curricular development (e.g., Hansen, 1995), as well as learning materials and environments (e.g., Nuffield, 1995). Here the key question is how the results of the research done in the previously mentioned areas (i.e., contents, skills, learning processes) may be translated into concrete instructional means and learning environments adapted to the needs and interests of different learners.

As an attempt to create a framework for the study of the above questions and the planning of learning situations and means according to different needs (e.g., age level, specialization level), we propose the model described in the following sections. The model comprises two main components. The first is the *technological problem-solving learning space*, for mapping the target cognitive goals and knowledge needs of different learners. The second component is the repertoire of *technological primitives*, referring to the set of skills and knowledge units and chunks assumed to be involved in TPS.

THE TECHNOLOGICAL PROBLEM SOLVING LEARNING SPACE

The first central component of our model is the learning space for technological-problem-solving knowledge and skills. The learning space is the framework within which actual learners – their cognitive goals, knowledge needs, cognitive functioning – can be situated. As such this represents an attempt to decompose the overly general definitions of 'technological expertise' and 'technological literacy' into more specific chunks of knowledge and skills according to the needs of different technology practitioners.

The learning space matrix is presented in Table I. First we can refer to a person's knowledge needs in terms of her or his functional goal regarding TPS. Correspondingly, cognitive goals, knowledge and skills, and their internal (cognitive) representations of these are considered. By internal representation we mean the cognitive constructs that store declarative and procedural knowledge, and a person's mental models of phenomena and systems (e.g., Norman, 1983; Kieras, 1988). In general terms we consider that each new level of practitioner (row in the table) includes features of previous levels, or is an elaboration and upgrading of these features. For example, we expect a person whose cognitive goal is to be able to repair or adapt an artifact, to be also competent in using it and to be aware of its basic features and functioning.

At one end of the scale we will refer to the *knowledgeable user*. A knowledgeable user of technology deals with TPS while having to express an opinion about a technology-related issue (e.g., conflictive aspects of the planned location of a powerful relay station), to choose an appropriate artifact for a specific need, and to be able to use and maintain it (e.g., substitute replacement parts). Knowledge and skills needed by this user include basic facts about the artificial world (e.g., terminology; common materials; measurement units – e.g., for electricity consumption, noise level, modem transmission rates); artifact operation and maintenance scripts; and knowledge required for technology-related value-judgment

TABLE I
Technological problem solving learning space

Function goal	Cognitive goal	Knowledge	International representation
Knowledgeable user	to be aware	– basic facts – social/economical/moral implications	– declarative knowledge-base
	to understand	– overall structure and functions	– phenomenal model
	to use	– operating scripts – maintenance scripts	– indexed script-repertoire
Problem solver (naïve to expert)	to repair	– device topography – functional configuration – assumptions about casual relationships	– qualitative causal model – buggy-but-working model – diagnostic model
	to adapt	– necessary/sufficient functions – debugging procedures – representational skills	– anticipatory model – solution-evaluation log
Practitioner	to build	– tooling skills – contextualized disciplinary knowledge – 'how-to' scripts	– pragmatic process-models – indexed local-solutions repertoire
Artisan	to craft	– ad-hoc design methods – generic disciplinary knowledge – normative knowledge	– proto-systemic runnable model – index generic-solutions repertoire
Expert	to design	– systematic design methodology – formal representational notations – formalized multi-disciplinary knowledge – knowledge-based heuristics	– systemic multiple-layer runnable model (declarative/procedural/qualitative) – formal anticipatory models

and decision-making. We may assume that this knowledge and these skills are internally represented as a declarative knowledge-base, a set of phenomenal models of components of the artificial environment, and a repertoire of indexed scripts for artifacts manipulation.

A *problem-solver* goes a step further than the knowledgeable user, coping with the challenge of restoring a faulty device to its normal functioning, amending a buggy process, or adapting either an artifact or a process to a new context or new usage constraints. The quality and complexity of the knowledge and of its internal representation will vary according to the degree of naïveté of the problem solver (for the sake of clarity in presentation and categorization, expert problem-solvers and professional troubleshooters are included in the 'Artisan' and 'Expert' categories at the expertise end of the scale). A naive problem-solver has to have at least a basic perception of the device topography and its functional configuration, generate assumptions about causal relationships among its parts, and be able to diagnose a fault and devise a repair plan. In addition, if he or she wants to adapt a solution to a new context, the naive problem solver has to know about necessary and sufficient conditions for the functioning of the system and have minimal planning skills and even basic representational skills. Internal representations may comprise, besides an augmented factual knowledge-base, qualitative (even partial or highly local) working models serving diagnostic or anticipatory purposes, and a log of evaluated attempts to repair or modify a device or a process (together with success/failure indexes).

Adapting existing solutions to new contexts or needs constitutes a transitional state between using technology and doing technology. However the *technology practitioner* is the first real 'technology maker' in our scale, dealing with the construction of new solutions. The population of practitioners in certain countries is rapidly growing due to the development of the 'do it yourself' culture, with all the required support (e.g., hardware stores, sophisticated tools and building kits, brief training courses, specialist magazines). The practitioner's knowledge develops as a result of motivated practice and goal-oriented learning. This knowledge comprises a rich set of tools and materials, manipulation skills, contextualized disciplinary knowledge in relevant areas (e.g., physical properties of materials) acquired mainly through experience and ad-hoc learning based on current-project needs, and a repertoire of 'how-to' scripts. We may assume the existence of a cognitive representation of this knowledge in the form of an indexed catalog of situated (proved) solutions and contextualized process-models.

Artisans or craftsmen are technology experts, our entry point to the realm of specialization. Solving technological problems is the essence of their professional activity. An obvious component of their knowledge is a comprehensive set of tools and materials handling skills in a given technological field. They own generic disciplinary knowledge which can be adapted and used in a variety of contexts and for different purposes. They know about the norms, standards and constraints within which the solutions should be generated. And they continuously devise ad-hoc design tactics and strategies, which gradually develop into a functional and useful design methodology. Internal representations of this knowledge take the form of an indexed catalog of generic and prototypical solutions, and runnable models of circumscribed solutions or artifacts.

Another level of expertise is that of the *professional designer*. The expert designer of technological solutions will rarely be involved (unlike the artisan) in the actual, physical making of all or most components of the solution. The designer's expertise comprises mastery of systematic design methods and strategies; formal representational notations and techniques (e.g., technical drawing, model building); formalized disciplinary knowledge (e.g., electronics or chemical processes) and multi-disciplinary knowledge (e.g., principles and laws of nature underlying a technological solution, or economic and social considerations for the design of a technological solution), and problem solving and design heuristics based on accumulated knowledge and experience. In addition to the cognitive constructs referred to at previous levels of technology knowledge and application, a key issue at this level is the ability to construct appropriate runnable mental models of the system under design. These models comprehend both structural and functional properties of the system and comprise multiple layers containing declarative, procedural and qualitative representations of the system. Moreover their mental running comes to support a variety of functions during the design process (e.g., mental exploration of alternatives, prospective evaluation of a solution, debugging).

The last three categories of 'technology makers' (i.e., practitioner, artisan and designer) correspond with Mitcham's (1994) reference to *bricoleurs*, *craftsmen* and *engineers*. The bricoleur – our practitioner – acts following intuition, using tools and materials at hand, with trial-and-error repetitive loops constituting her or his main methodological routine. At the other extreme the engineer – our expert designer – constructs by means of systematic analysis and synthesis; she or he is not limited but rather challenged by contextual constraints. Mitcham considers craftsmanship or artisan making as intermediate between bricolage and engineering: it is no longer strongly tied to circumstantial factors or constraints, but nor does it proceed solely on the basis of successful trials. On the other hand, craftsmanship does not follow the conceptual methodology of engineered problem solutions. Craft action is based on a systematic set of techniques, material-handling procedures and solution templates, but it is still committed to contextual features (e.g., naturally sound materials or solutions) and the complete-crafted solution (e.g., it generates one complete solution – there is no mass production or division of labor).

TECHNOLOGICAL PROBLEM SOLVING PRIMITIVES REPERTOIRE

The second main component of our proposed framework is the *Technological-Problem-Solving Primitives* repertoire. The *primitives* are the basic units or building-blocks of the problem solving process. This component of the model is an attempt to map the various levels of skills, and knowledge units and chunks, assumed to be involved in TPS.

The primitives were classed into four categories (Figure 2): Technological Problem Solving Rudiments (T-ruds), Models (T-models), Methods (T-methods) and Metaknowledge (T-metas).

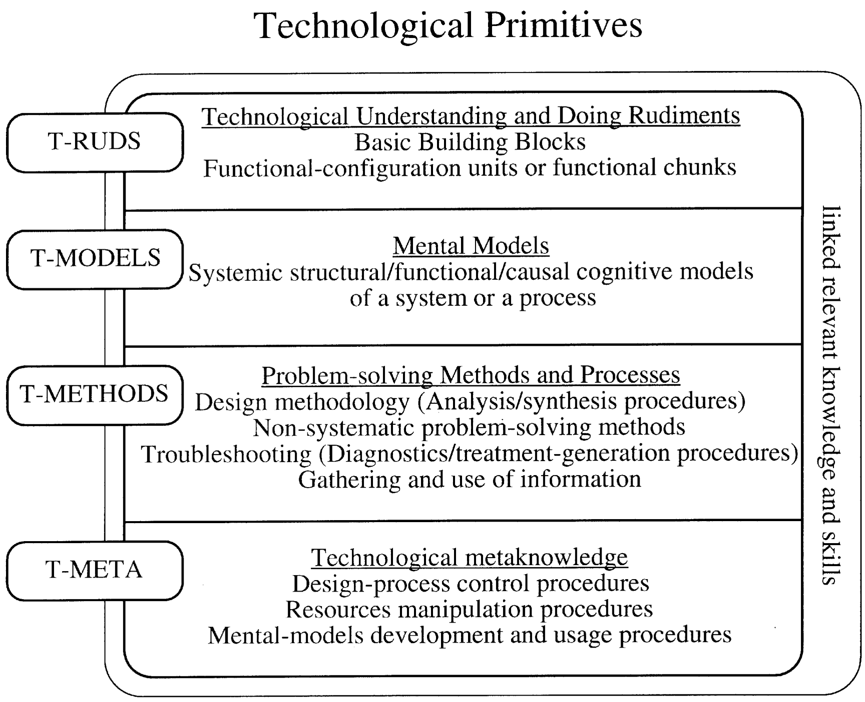

Figure 2. Technological primitives.

TPS Rudiments (T-ruds)

These are the basic building blocks of technology related performance. They consist of a wide range of knowledge units which are activated in different situations, e.g., for operating devices, using tools, building models, or programming a computer. Rudiments may be single pieces of knowledge but they may also be chunks built out of these single pieces, composite units for solving technology-related problems. In this case they may take the form of scripts (Schank & Abelson, 1977), knowledge aggregates, or clusters of skills. Rudiments include declarative knowledge (e.g., properties of materials, measurement units, types and functions of Lego building blocks), and procedural knowledge (e.g., how to operate a toaster oven, how to assemble Lego bricks, silk-screen printing). The following are but a few examples of T-ruds for different practitioners in the learning space matrix.

(a) On the Knowledgeable-user level we may consider the 'Generic-knob-functioning' building block. At the generic level a user perceives a knob as an input device of a system, and recognizes the correspondence between its clockwise rotation and an increase in the input value (e.g., increase in a receiver's volume, an oven temperature, a delay in the activation of a device controlled by a timer). The user is also able to accommodate exceptional versions of the knob's generic functioning, as in the refrigerator (which increases in cold as opposed to the increase in heat in the oven's knob) or moreover as in a sound-system control unit (there is no reason to think in terms of an 'increase' when turning the knob to choose from among the CD, tape, tuner or TV set input to the system).

(b) On the practitioner level a young student applies the generic-knob model to program the functioning of a Lego bricks fan at different velocities according to the position of a rotation sensor based control knob.

(c) Typical examples of T-ruds are also basic model-construction rudiments. For example O., a seven year old girl, worked for a while on solving the problem of fixing vertical bars on her Lego model of a swing. A suggestion was made to her to explore the small plastic pins in the box to get a perpendicular juncture between two Lego blocks (Figure 3a). From now on the perpendicular-juncture building rudiment was adopted and adapted by O. for different components of her playground, as in the spinning-barrel device (Figure 3b).

T-models

The problem solver's mental model of a target technological system or a problem situation is a key factor in the problem solving process, whether the person is trying to understand the system (diSessa, 1983; Hegarty, 1988; Simons & Keil, 1995), predict its behavior (Williams et al., 1983; de Kleer & Brown, 1983), operate it (Kieras & Bovair, 1984; Stigler, 1984), repair

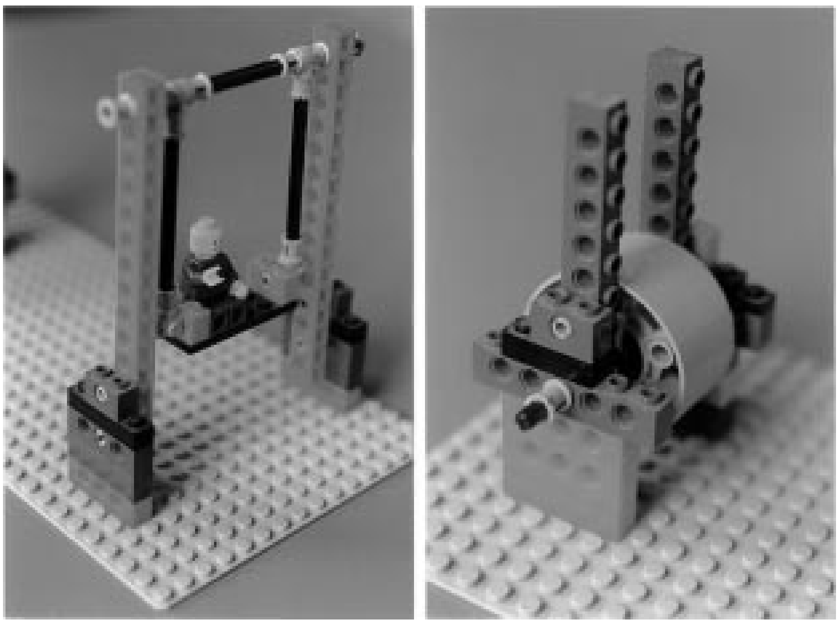

Figure 3. Model-construction rudiments.

it (Lajoie & Lesgold, 1989; Sanderson & Murtagh, 1990), or design a new one (White & Frederiksen, 1986; Moray & Reeves, 1987; Mioduser et al., 1996).

In considering these models as primitives or resourceful units for problem solving, we should reflect on a number of crucial issues: their structure, the methods by which they are constructed, the way they evolve, the way they are retrieved and used, the different kinds of models, and the correspondence between models' properties and individual differences among problem solvers (e.g., age level, level of expertise).

For example, de Kleer and Brown (1983) suggested that a person models a technological system through the following process: (a) an initial internal representation of the structural configuration of the system, called 'device topology' is created; (b) this is complemented by a process called 'envisioning' – by which the system's functional configuration is inferred from its structure; and (c) this results in a particular causal model or 'runnable mental model' of the system. At the device topology level, the model has several constituents: parts (e.g., energy source, valves, clapper), conduits connecting the parts (e.g., pipes, wires), and 'stuff' flowing through the conduits (e.g., oil, electrons, water).

Regarding the evolution of models, we may cite as an example our study of students' designing of simple control systems (Mioduser et al., 1996). Students worked first on a series of learning tasks analyzing examples of opening-closing automated systems from the natural world (e.g., the Venus flytrap, the epiglottis in the human throat) and the artificial world (e.g., automatic doors or faucets). Then they were asked to design and build a computer controlled opening-closing mechanism. Based on different types of data (e.g., written descriptions, drawings, verbal descriptions, videotaped working sessions) we characterized the students' models of the systems in terms of a 'structural differentiation' and 'functional specialization' evolution. As shown in Figure 4, model (a) is an undifferentiated general input/output model, while at the other end, model (d) represents a complete causal model indicating input/output relationships between the Operating Unit (OU) and the Control Unit (CU), and a clear differentiation among data collection functions, decision making functions, and operation functions.

These models served as primitive frameworks guiding the analysis and design tasks, but at the same time they evolved through repeated experience, thus becoming more powerful primitive tools in terms of optimization of solutions, search for alternative solutions, and explanation or prediction of the functioning of the system.

T-methods

TPS is commonly identified in the literature as well as in teaching practice with what is called 'the design process' (e.g., McCormick et al., 1994). In this section we want to elaborate on the need to (a) clarify the signifi-

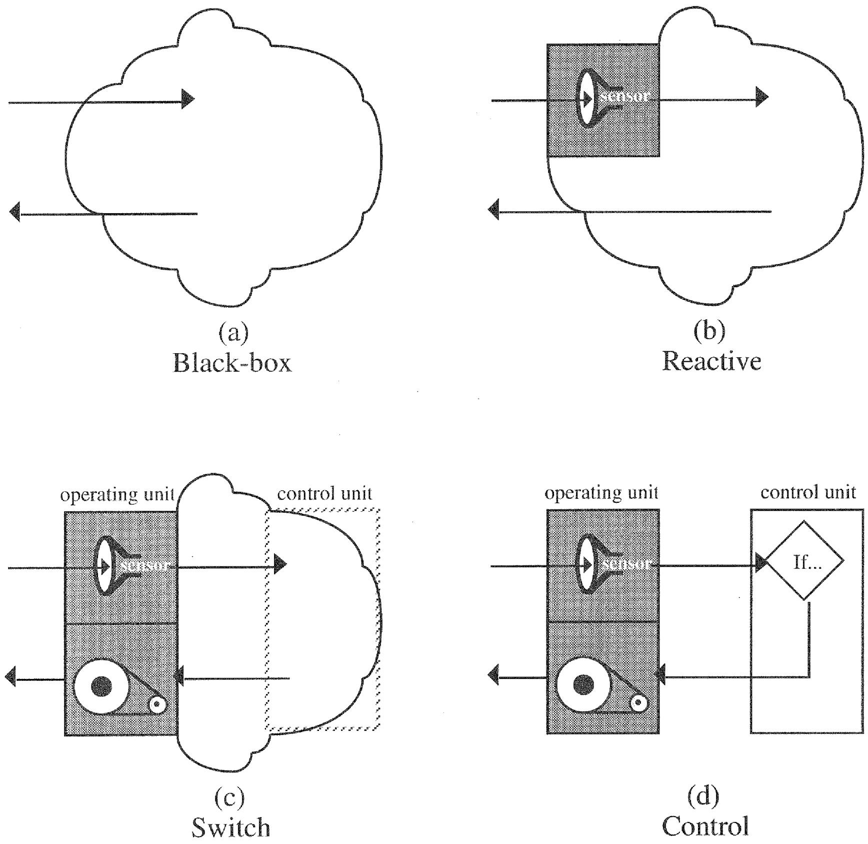

Figure 4. Mental models of a simple control system.

cance and role of this design process within learning and teaching situations (as distinguished from engineering or professional design situations), and (b) decompose the overall process into its methodological components which qualify as methodological primitives for TPS.

Within the professional community dealing with design (e.g., engineers) a certain degree of consensus exists regarding the overall definition and stages of the design process: identification of problems and diagnosis of needs, through a series of loops at which solutions are conceived, explored and evaluated until a suitable answer is found and then instantiated (Bucciarelli, 1996). But beyond that consensus the process is usually interpreted in flexible terms, leaving space for alternative configurations of the process stages, according to varied parameters, e.g., the nature of the problem or the technological field, the composition of the design team, time or financial constraints, or level of novelty of the problem or the desired solution (e.g., see Bucciarelli, 1996, for detailed descriptions of the dynamics of problem solving processes by engineers).

Paradoxically, this flexibility has been lost when the design process model was redefined for educational purposes. Its reformulation by educators, in recent theoretical proposals as well as in curriculum development efforts, has somehow resulted in a rigid sequence (e.g., DES, 1990). A generic description of this sequence will include the following steps: (a) identifying needs and problems, (b) investigating and generating a design brief, (c) generating and exploring alternative solutions, (d) choosing a solution (and eventually modeling it), (e) building the chosen solution and (f) evaluating the outcome. The process proceeds stage-wise and a 'return path' from the evaluation stage to all previous stages (revision loop in light of outcome evaluation) is usually suggested.

Does this clearly defined sequential model correspond to real-life design processes? Or does this type of educational model run the risk of offering a too rigid, reductive representation of a rich, fluid and creative process? An even more important question is: Does this model correspond to real-life design processes done *by students*? Initial research results on children's design activities suggest that they design in many different ways. For example, McCormik et al. (1994) found a substantial mismatch between students' perception of the design process and that of the teachers and curriculum developers. Far from keeping in mind the process as a whole and from referring to it while solving problems at specific stages, the students perceived each step as quite an independent task in itself. Researchers noted that specifications made at earlier (planning) stages were ignored at later (building and evaluation) stages, suggesting a lack of perception of the continuity and organicity of the process.

Our own observations instructed us about the particular importance of the student's own re-formulation of the design-stages sequence, and of the task sequence within each stage, guided by her or his perceived needs at each step. Highly contextual data as perceived by the student fueled a chain of specific decisions which resulted in nested action-loops involving one or more stages of the design process. Let us consider the example of a group of junior-high students who were asked to build a working Lego model of a technological device (e.g., car, helicopter, machine). We focused our analysis on a particular section of the process, starting when the students finished building their first version of the model. From that point on they entered a series of loops which we called 'evaluation-modification-loops', until they made the decision that the task was completed. Figure 5 shows a model of the decision-making space for this section of the process, based on our observations of the students' performance (a detailed description of this model and the study sustaining its formulation is beyond the scope of this paper – see Kiperman, 1997; Mioduser & Kiperman, in press).

Once the first version of the device was constructed, we found that all students proceeded to a plainly technical step: to connect the device to the interface box and check if it actually worked. If it worked, an *evaluation-against-original-goal* and improvement track was followed. If it did not work a *fault-diagnosis* and (eventually) *goal-modification* track was

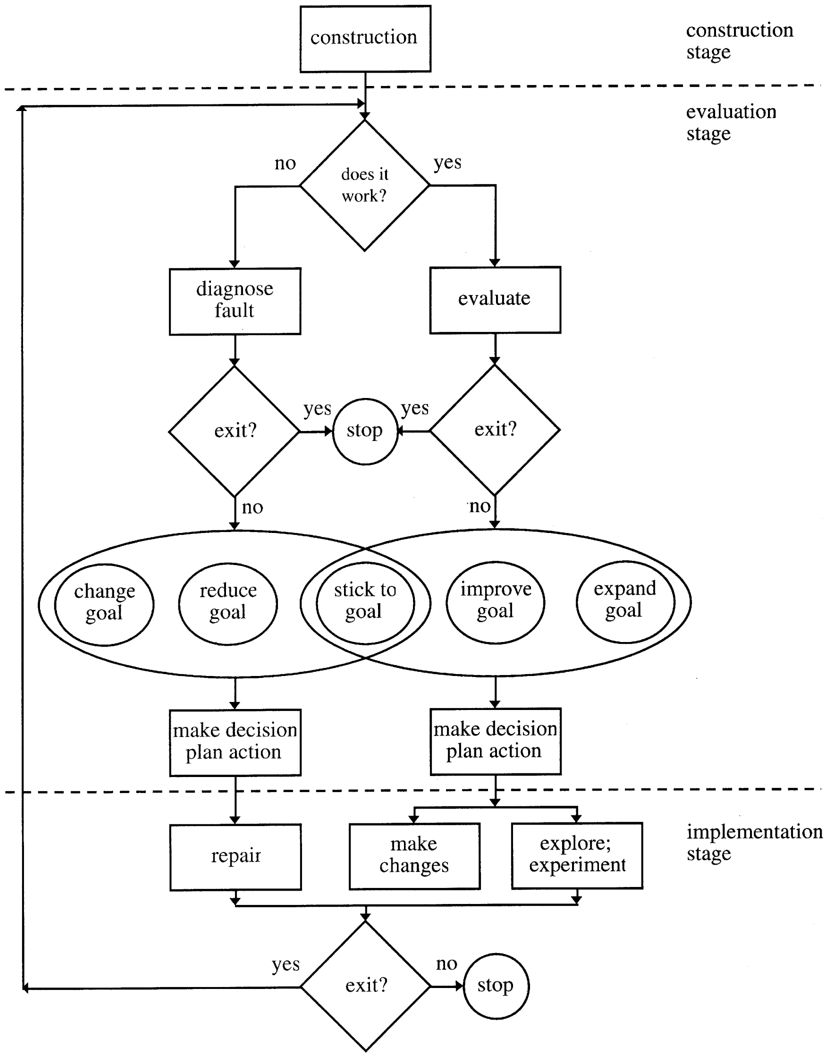

Figure 5. Evaluation-modification-loops model.

followed. In both eases the (faulty or functioning) device was evaluated with reference to the original goal, leading to decisions which ranged from sticking to the goal at one end, up to abandoning and re-formulating it in the other. Depending on the decision (e.g., same goal, expanded goal, new goal, etc.) actions were planned and implemented, and a new loop started (from the technical checking stage), until the student decided that the work had come to an end. In this example we were able to identify different design functions (e.g., exploration, goal formulation, evaluation) which were interwoven within the different stages of the evaluation-modification loop, rather than appearing as linear or sequential fixed steps.

These and other studies provide two lines of thought as to the definition of methodological primitives for the teaching and learning of TPS. The first refers to the prospect of identifying student strategies regarding particular stages within the design process (e.g., the different paths in the 'evaluation-modification-loops'). The second illuminates the problem solving or design process as a whole. We would suggest that a better formulation of the process in terms of what the students actually do, as an alternative to the sequential model, is offered by the model shown in Figure 6. We suggest that students are engaged in four main activities while designing a technological solution, namely, (a) problem identification and definition of goals and constraints for the solution; (b) exploration of ideas, materials, energy forms, information forms, mechanisms and processes; (e) construction, and (d) evaluation. All four blocks are interconnected and any path or looping within this space may represent the problem solving process of an individual or a group. Our observations indicate that while all four activities are always to some extent part of the process, the paths chosen and the number of times each activity recurs are precisely what characterizes the particular problem solving performance of an individual student.

In addition to the above elaboration of the design process as the core methodological tool for TPS, we also observed a need to expand the methodological primitives repertoire to include additional methods. Some relevant candidates could be for example information retrieval methods (e.g., serving the 'goal and constraints definition stage' or the 'exploration stage'), sys-

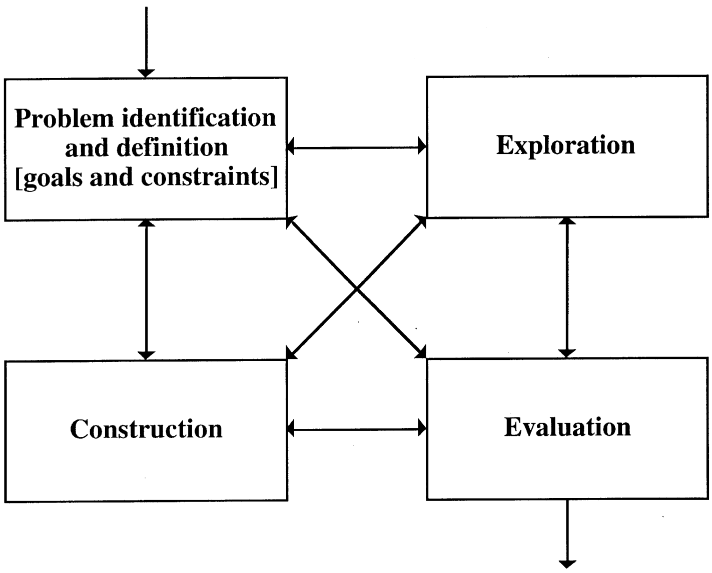

Figure 6. Four blocks design.

tematic data gathering, model building, troubleshooting, or debugging methods. All of these are by themselves methodological chunks, which when decomposed reveal the methodological units they are built of.

T-metas

Meta-level primitives concern the ways the learner uses primitives from the previous levels (e.g., rudiments, models, methods) and controls the problem-solving process. They may be referred to as the metacognitive aspects of the TPS process. Metacognition refers to the learner's knowledge about her/his own cognition and to the control and regulation of her/his own cognitive actions (Flavell, 1976; Brown & DeLoache, 1978; Slife et al., 1985). For the TPS process, two sets of primitives are of relevance: those related to the meta-level properties of the cognitive processes involved in solving technological problems, and those related to the control and regulation of the process.

Let us consider an example of the first set. The following meta-level properties have been suggested by theorists and researchers to characterize technological thinking, e.g.: analytic; synthetic; goal-oriented; anticipatory; exploratory; reflective; case-based; inventive; serendipitous; systemic (e.g., Mitcham, 1994; Chen & Strupp, 1993). It can be assumed that corresponding meta-level primitives should be activated whenever a learner wants to mold a cognitive path to fit a desired property. For example, regarding serendipity ('the art of finding things you are not looking for') Norbelt Wiener suggests that in the light of 'surplus' results of a systematic research process, the question 'Now that I have come to a result, what problems have I solved?' (a meta-level primitive in our terms) is of great value, specially when it comes to particular engineering problems (1994, pp 21–22).

The second set of primitives relates to the learners' control over the process. Some examples are shown in Table II. For example, exploration activities may proceed in a tree-like fashion leading (branching) from one issue to another. In this case the [stop exploration-routine?] procedure is highly relevant to making the decision as to when enough searching has been done, so that is possible now to proceed to the next step. An example at another level of primitives relates to the mechanism by which a new primitive (e.g., a new successfully used building rudiment) is consciously indexed (and eventually stored) for further retrieval.

FINAL REMARKS

Technology education is undergoing extraordinary changes in the last decade. Educators and educational policy makers have become aware of the importance of *technology* in the basic formation of today's educated person. At the same time rapid and continuous change in technology itself poses

TABLE II
Example of process control meta-level primitives

Design-process control procedures	
e.g.,	[stop exploration-routine] decision
	[expand goal] procedure
Resources manipulation procedures	
e.g.,	[retrieval-adaptation-use] of prototypical T-prims
	[formalization-indexation-storage] of new T-prims
Mental-model development procedures	
e.g.,	[differentiation-specialization] processes
	[prototypical-model retrieval-and-instantiation]
Serendipitous-events monitoring and treatment	
e.g.,	reverse-engineering for unexpected outcome

serious challenges in regards education and training in this field. Thus technology related contents, skills and teaching strategies for all specialization levels are being reconsidered and redefined for educating both the technologically literate person and the expert practitioner.

Within the discipline, *technological problem-solving* or the process and skills by which people design solutions, constitutes a key component. Yet the study of teaching, learning, and people's use of these skills and processes is only in its initial stages. Aiming to support the identification of problematic issues and the generation of relevant research questions, as well as to frame instructional requirements according to the needs of different target populations, we proposed the TPS model. This model serves us as a framework of our current research projects. For example, regarding cognitive models of technological systems, one of our studies focuses on learners' perceptions of interactions in a system (both internal – among subsystem's – and external – among a system and its environment-), as well as the perception of the relationship between these interactions and the systems functioning. Another example, this time at the instructional level, is our effort to create a control programming environment built upon a progression of cognitive models of control functions and procedures.

As evident in the previous sections, part of the cells in the proposed framework are still in skeletal form awaiting more detailed definition. It is our hope that the growing body of knowledge generated by the technology education community will contribute to the gradual completion of the model components.

REFERENCES

Bensen, J.: 1995, 'A Context for Technology Education', in G. E. Martin (ed.), *Foundations of Technology Education, 44th Yearbook of the Council on Technology Teacher Education,* Glencoe/McGraw-Hill, Peoria, pp. 1–24.

Brown, A. & DeLoache, J.: 1978, 'Skills, Plans and Self-Regulation', in R. Siegler (ed.), *Children's Thinking: What Develops?*, Erlbaum, Hillsdale, NJ.

Bucciarelli, L.: 1996, *Designing Engineers*, MIT Press, Cambridge, MA.

Chen, D. & Strupp, W.: 1993, 'General Systems Theory: Towards a Conceptual Framework for Science and Technology Education for All', *Journal of Science Education and Technology* **2**(3), 447–459.

DES: 1990, *Technology in the National Curriculumum*, Department of Education and Science and the Welsh Office, HMSO, London.

de Kleer, J. & Brown, J. S.: 1983, 'Assumptions and Ambiguities in Mechanistic Mental Models', in D. Gentner & A. L. Stevens (eds.), *Mental Models*, Erlbaum, Hillsdale, NJ.

de Vries, M.: 1996, Technology Education: Beyond the "Technology as Applied Science" Paradigm', *Journal of Technology Education* **8**(1) (no page numbers (electronic version)).

di Sessa, A.: 1983, 'Phenomenology and the Evolution of Intuitions', in D. Gentner & A. Stevens (eds.), *Mental Models*, Erlbaum, Hillsdale, NJ.

Dreyfuss, H. L. & Dreyfuss, S. E.: 1986, *Mind Over Machine. The Power of Human Intuition and Expertise in the Era of the Computer*, The Free Press, New York, NY.

Dyrenfurth, M. J.: 1991, 'Technology Literacy Synthesized', in M. J. Dyrenfurth & M. Kozak (eds.), *Technological Literacy, 40th Yearbook of the Council on Technology Teacher Education*, Glencoe Division, Macmillan/McGraw-Hill, Peoria, IL, pp. 138–183.

Flavell, J.: 1976, 'Metacognitive Aspects of Problem Solving', in L. Resnick (ed.), *The Nature of Intelligence*, Erlbaum, Hillsdale, NJ.

Gardner, P.: 1995, 'The Relationship between Technology and Science: Some Historical and Philosophical Reflections, Part II', *International Journal of Technology and Design Education* **5**(1), 1–33.

Gropius, W.: 1975, 'The Theory and Organization of the Bauhaus', in H. Bayer, W. Gropius & I. Gropius (eds.), *Bauhaus*, Secker & Warburg, London.

Grubb, W.: 1996, 'The "New Vocationalism" in the United States: Returning to John Dewey', *Educational Philosophy and Theory* **28**(1), 1–23.

Hansell, K.: 1995, 'Reflection on Technology Education. A Curriculum Model, Research and Development', *International Journal of Technology and Design Education* **5**(1), 35–50.

Hegarty, M.: 1988, Mental Models of Mechanical Systems: Individual Differences in Qualitative and Quantitative Reasoning', *Cognitive Psychology* **20**(2), 191–236.

Ihde, D.: 19901, *Technology and the Lifeworld – From Garden to Earth*, Indiana University Press, Bloomington.

Ihde, D.: 1993, *Philosophy of Technology: An Introduction*, Paragon House, New York, NY.

Johnson, S. D., Foster W. T. & Satchwell, R.: 1989, July, *Sophisticated Technology, the Workforce, and Vocational Education*, Department of Adult, Vocational and Technical Education, Illinois State Board of Education, Springfield, IL.

Kieras, D.: 1988, 'What Mental Model Should be Taught: Choosing Instructional Content for Complex Engineered Systems', in J. Psotka, L. D. Massey & S. A. Mutter (eds.), *Intelligent Tutoring Systems – Lessons Learned*, Erlbaum, Hillsdale, NJ.

Kieras, D. & Bovair, S., 1984, 'The Role of a Mental Model in Learning to Operate a Device', *Cognitive Science* **8**(3), 255–273.

Kiperman, D.: 1997, *Evaluation-Modification Loops in Students' Design Process of Technological Devices*. Unpublished M.A. Thesis, Tel-Aviv University, School of Education.

Lajoie, S. & Lesgold, A.: 1989, 'Apprenticeship Training in the Workplace: Computer-Coached Practice Environment as a New Form of Apprenticeship', *Machine-Mediated Learning* **3**(1), 7–28.

Lesgold, A., Lajoie, S., Bunzo, M. & Eggan G.: 1992, 'Sherlock: A Coached Practice Environment for an Electronics Troubleshooting Job', in J. Larkin & R. Chabay (eds.), *Computer Assisted Instruction and Intelligent Tutoring Systems: Shared Goals and Complementary Approaches*, Erlbaum, Hillsdale, NJ, pp. 201–238.

Levin, I. & Mioduser, D.: 1996, 'A Multiple-Constructs Framework for Teaching Control Concepts', *IEEE Transactions on Education* **39**(4), 488–496.

McCormick, R., Murphy, P. & Henessy, S.: 1994, 'Problem-Solving Process in Technology Education: A Pilot Study', *International Journal of Technology and Design Education* **4**, 5–34.

Mino, M.: 1995, 'A Futuristic Vision Technology for all Americans', *The Technology Teacher* **55**(4), 3–4.

Mioduser, D.: 1998, *Technological Problem Solving – Student Book*, ORT Israel (Hebrew), Tel-Aviv.

Mioduser, D., Venezky, R. L. & Gong, B.: 1996, 'Student's Perceptions and Designs of Simple Control Systems', *Computers in Human Behavior* **12**(3), 363–388.

Mioduser, D. & Kiperman, D.: In press, *Evaluation-Modification Loops in Children's Technological Problem-Solving*.

Mitcham, C.: 1994, Thinking Through Technology – *The Path Between Engineering and Philosophy*, The University of Chicago Press, Chicago, IL.

Moray, N. & Reeves, T.: 1987, 'What Operators Learn about Complex Systems', in *Proceedings of the 1987 IEEE International Conference on Man Systems and Cybernetics*, IEEE, New York.

National Commission on Excellence in Education: 1983, *A Nation at Risk: The Imperative for Educational Reform*, US Department of Education, Washington, DC.

National Research Council: 1996, *National Science Education Standards*, National Academy Press, Washington, DC.

Norman, D.: 1983, 'Some Observations in Mental Models', in D. Gentner & A. L. Stevens (eds.), *Mental Models*, Erlbaum, Hillsdale, NJ.

Nuffield: 1995, *Nuffield Design and Technology – Students Book*, Longman, London.

Raizen, S. A., Sellwood, P., Todd, R. D. & Vickers, M.: 1995, *Technology Education in the Classroom: Understanding the Designed World*, Jossey-Bass Publishers, San Francisco.

Sanderson, P. & Murtagh, J.: 1990, 'Predicting Fault Diagnosis Performance: Why Are Some Bugs Hard to Find', *IEEE Transactions of Man Systems and Cybernetics* **20**(1), 274–283.

Savage, E. & Sterry, L.: 1990, 'A Conceptual Framework for Technology Education', *The Technology Teacher* (September/October), 6–11.

Schank, R. & Abelson, R.: 1977, *Scripts, Plans, Goals, and Understanding*, Erlbaum, Hillsdale, NJ.

Simon, H.: 1981, *The Sciences of the Artificial*, MIT Press, Cambridge, MA.

Simons, D. & Keil, F.: 1995, 'An Abstract to Concrete Shift in the Development of Biological Thought: The Inside Story', *Cognition* **56**, 129–163.

Slife, B., Weiss, J. & Bell, T.: 1985, 'Separability of Metacognition And Cognition: Problem Solving in Learning Disabled and Regular Students', *Journal of Educational Psychology* **77**(4), 437–445.

Snyder, J. & Hales, J.: 1981, *Jackson's Mill Industrial Arts Curriculum Theory*, West Virginia Department of Education, Charleston.

Stevens, E.: 1995, *The Grammar of the Machine*, Yale University Press, New Haven.

Stigler, J.: 1984, '"Mental abacus": The Effect of Abacus Training on Chinese Children's Mental Calculation', *Cognitive Psychology* **16**, 145–176.

Vincenti, W.: 1990, *What Engineers Know and How They Know It*, John Hopkins University Press, Baltimore, MA.

Vos, E.: 1996, 'New Media Poetry – Theory and Strategies', *Visible Language* **30**(2), 214–233.

White, B. & Frederiksen, J.: 1986, *Progressions of Qualitative Models as a Foundation for Intelligent Learning Environments*, Report No. 6277, BBN Laboratories Inc., Cambridge, MA.

Wiener, N.: 1994, *Invention – The Care and Feeding of Ideas*, MIT Press, Cambridge, MA.

Williams, M., Holland, J. & Stevens, A.: 1983, Human Reasoning about a Simple Physical System', in D. Gentner & A. L. Stevens (eds.), *Mental Models*, Erlbaum, Hillsdale, NJ.

Wu, T., Custer, R. & Dyrenfurth, M.: 1996, 'Technological and Personal Problem Solving Styles: Is There a Difference?', *Journal of Technology Education* **7**(2) (no page numbers (electronic version)).