

Understanding the Importance of Intermediate Representations in Engineering Problem-Solving

Durward K. Sobek, II
Montana State University

Abstract

This paper describes an emerging theory on the role that representations play in engineering problem solving. Modern cognitive psychology has shown that not only do problem solvers use different representations to store information and ideas, the representation itself influences the problem-solvers' solution approach. I extend this notion to the engineering domain, and illustrate it with an example from programmable logic controller (PLC) programming. These ideas have important implications for how educators can help students develop effective problem-solving skills.

1. Introduction

Many students, when faced with an engineering problem, attempt to proceed directly from problem statement to solution. Research on the problem-solving strategies has shown that expert problem-solvers spend more time and effort on problem understanding (or problem definition) than do novices, and this enables them to arrive at better solutions,¹ and often more quickly.² However, exhorting novices to “understand the problem better” before trying to solve is often futile.

A stream of research in cognitive science has studied the very important role that external representations play in problem solving. Problem-solvers use representations of problem spaces (written descriptions, diagrams, flow charts, mathematical equations) and of solution spaces (requirement lists, sketches, schematics, programming code, quantified entities) in their solution approaches.^{3,4,5} A number of studies have concluded that the expert problem-solver is often the person “with a better map”—that is, a better representation.⁶

I hypothesize that engineering education can be enhanced with the introduction of appropriate and effective *intermediate representations* that help students bridge the cognitive gap between problem statement and potential solutions. Intermediate representations are carefully designed tools/techniques to help the problem solver think through the problem space more deeply, systematically, and carefully, and generate a better understanding of the problem parameters and relations among them.

The following section outlines the theoretical arguments for the role representations play in problem solving, and defines the term “intermediate representation.” The next section illustrates the concept using the case of programmable logic controller programming. The paper concludes

with a discussion of the characteristics of a “good” intermediate representation, encourages future research into developing effective representations for specific problem domains, and strongly suggests that these representations should be an explicit part of engineering curricula.

2. Cognition, Representation, and Problem-Solving

Approaches to cognitive theory for the better part of the 20th Century view human cognition largely as an internal process.⁷ Humans take in information and create an internal model of the external environment, often termed “encoding.” They then process the information through some sort of mental computation. Some researchers have divided the mental processes into subprocesses, such as retrieval (fetching information stored in memory), mapping (comparing new information to stored information), and evaluation (finding deficiencies in the new and/or stored models).⁸ The results of internal processing are then externalized through some sort of decoding process, and the cycle begins again. This common view formed the basis for much cognitive science investigation of the last century, and continues as an influential model to this day.

However, as cognitive psychologists began to test the explanatory value of the theoretical models, they found phenomena that the theory could not explain well. This has paved the way for a new theoretical framework to emerge. *Sociocultural theory*, initially developed by Soviet psychologist Lev Vygotsky and furthered by a number of his followers, has received increasing attention in the cognitive science community over the last couple of decades. It provides a richer, multifaceted framework for understanding human development and cognition that addresses many of the deficiencies of earlier theory.⁹

Sociocultural theory emphasizes that humans do not construct knowledge (the foundation of human cognition) through individual, internal processes. Rather, the theory posits that knowledge emerges from the interaction of social and individual processes, that knowledge is co-constructed through this interdependence.¹⁰ One of the centerpieces of the sociocultural framework is the importance of semiotic mediation in human development.¹⁰

Semiotic mediation refers to the use of language, counting systems, mnemonics, symbol systems, writing, diagrams, and other signs and symbols in human cognition. Vygotsky and his followers argue that knowledge is not internalized directly, but through the use of so-called “psychological tools” appropriated during the course of intellectual development (also called “cultural tools”). These tools are not developed by the individual in isolation, but rather, like language, emerge as products of sociocultural evolution. A psychological tool in sociocultural discourse is virtually any instrument connected with conceptual thought—calendars, the computer, maps, mechanical drawings, and works of art to name a few. One’s mental functioning is tied to the cultural and social settings in which the individual masters the tools s/he uses to help reason through a situation. The tools become carriers of the sociocultural patterns of knowledge, which the individual actively engages and at the same time influences.

Some recent research on problem solving seems to follow a similar vein. Zhang distinguishes between external and internal representations of knowledge.⁷ External representations are knowledge and structure in the environment (or external to the mind), such as physical symbols, objects, relational configurations. The sociocultural theorist would call many if not all of these psychological or cultural tools. Internal representations are knowledge and structure stored in

memory, such as propositions, deduction, and schema. Traditional approaches to cognition interpret external representations as simple extensions of memory, placeholders for ideas and information. The sociocultural view, however, says that the external and internal representations interact in the creation of new knowledge through a process called mediation. Mediation means one's internal mental processes act through the external representations used, and are therefore fundamentally affected by them.

Zhang and Norman provide a poignant example of how the form of external representation changes cognition.¹¹ They created several formulations of the tic-tac-toe, including the standard form of X's and O's and a numerical form (add numbers 1-9 to get 15). Study participants were quick to recognize the common "middle first" strategy to tic-tac-toe when using the X's and O's representation, but often did not discover the counterpart of this strategy for the other formulations until many iterations later, if at all. The authors claim that the representational form affects the problem-solving approach, even though the underlying problem structure was the same.

When we look at engineering problems, we see solutions and solution approaches rife with psychological tools. A typical solution to a simple statics problem, for example, involves a free body diagram followed by algebraic expressions that sum forces and moments followed by numerical calculations. Likewise circuit diagrams are used for electronics problems, cash flow diagrams for economic problems, and balance equations for chemical problems. Each of these is a psychological or cultural tool, developed over time to help solve certain kinds of problems. We can, therefore, view many engineering problem-solving processes as series of representational transformations that take us from problem to solution.

Engineering students are taught in many of their classes to get problems into standard formulations (or representational forms), then apply a set of standard procedures to get an answer. For example, in linear programming problems, the most difficult task is setting up the tableau correctly. Once this is done, applying the Simplex method is a straightforward step-by-step procedure easily mastered by most students. The difficulty and challenge in problem solving is not usually following the procedures, but rather getting from problem statement to a standard formulation. Sociocultural theory tells us that focusing on appropriate semiotic mediation can help. I term tools that bridge the gap between problem and standard formulation, "intermediate representations."

3. Representation in PLC Programming

An example will help illustrate several of the concepts just described. Programmable Logic Controllers (PLC's) are special purpose microcomputers designed for controlling discrete processes. They use a unique programming language called ladder logic. Ladder logic evolved from relay logic developed in the days of manufacturing automation before computers were widely available. Implementation using a microcomputer expands PLC capability to include timers, counters, analog-digital conversion, and many other advanced functions.¹²

Figure 1 describes a simple control problem and shows the solution in ladder logic. An easy way to interpret a ladder diagram is to think of the left rail as a positive electrical terminal and the right rail as ground. Each rung has only one output. The inputs are logically “closed” (slash mark) or “open” (no slash), and can be arranged in series (logical AND) or in parallel (logical OR). If, when the microcomputer scans the inputs, the rung’s logic is true, then the output is activated; otherwise, it is not activated. A ladder diagram can contain multiple rungs, inputs may be used more than once, and outputs can be inputs making interlock situations possible.

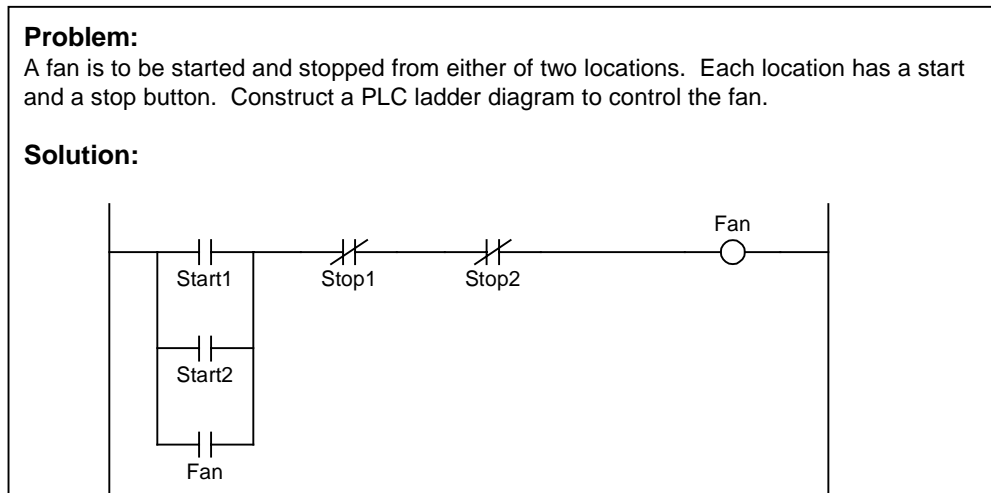


FIGURE 1: PLC PROBLEM AND SOLUTION

The solution approach depicted in Figure 1 implies that the problem-solver was able to develop a solution directly from the problem statement. This may be possible if the programmer is fluent in ladder logic and the problem is sufficiently simple. However, students learning PLC programming are barely familiar with ladder logic. They typically find drawing a logic diagram directly a cognitively taxing exercise, even for simple problems. Logically incorrect solutions often result; and if correct, the process often takes a surprisingly long time.

I introduced my students to an intermediate step I call *I/O mapping*. The technique is to draw a diagram of the system that includes all inputs and outputs. Then, the student reads the problem statement phrase by phrase and draws arrows from every device that affects another to the affected device. Finally, the student constructs a rung for each output in the system. The rung should include an input for each incoming arrow. To do this last step, creating a Boolean algebraic expression is often helpful. Boolean algebra represents the logical relationships of the different rung elements, from which the ladder diagram can be readily derived. (My students gain some familiarity with Boolean algebra earlier in the course.)

Figure 2 illustrates this alternative approach. From the problem statement, a simple diagram of a fan and two control switches is drawn (A). Then, since the problem statement says that the fan is to be controlled from both locations, arrows are drawn from the control switches to the fan. Next, a Boolean expression (B) describes the logic for the only output in the system, the fan. In English, the Boolean expression reads, “The fan is true (or on) if Start1 or Start2 is on, or if the fan is on, and Stop1 is off and Stop2 is off.” From the algebra, the ladder logic (C) is straightforward.

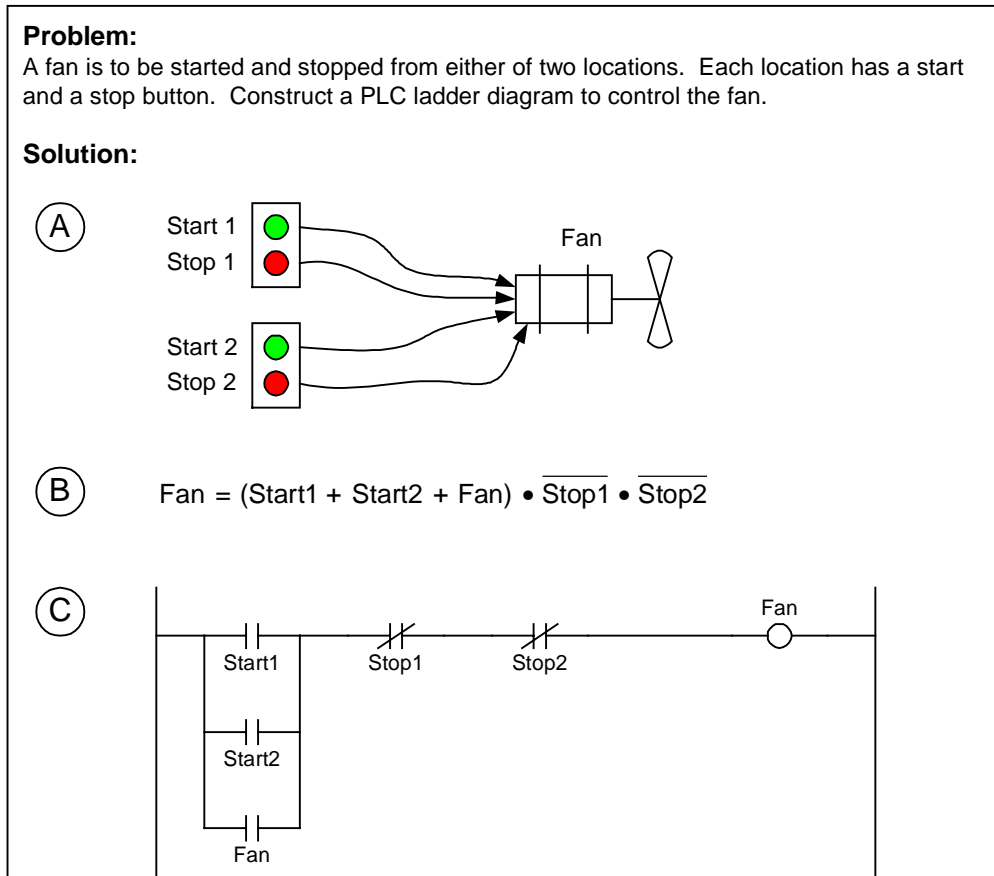


FIGURE 2: PLC PROBLEM AND SOLUTION USING INTERMEDIATE REPRESENTATIONS

When students use the intermediate representations, they tend to solve PLC problems more quickly and more accurately, even though the approach incorporates more steps than the direct solution approach. This observation has been verified and documented empirically through a classroom experiment, with approximately 60% of students improving in problem solving efficiency and/or solution quality without decreasing the other (see a companion paper for a more detailed description of the empirical results¹³).

What explains the apparent effectiveness of the intermediate steps? First, reformulating the problem statement into a different form helps the students think about the problem statement more carefully than s/he might otherwise.¹⁴ Simply taking in the information as presented and trying to process it is difficult. A psychological tool like the I/O map helps the student identify the different system elements through the physical act of drawing each element. Having to choose a physical location on the diagram for each element initiates the understanding of interactions between elements. The direction of the arrows helps identify inputs versus outputs. So we see a very strong interaction between the external representation (the I/O map) and the internal mental processing required to solve the problem.

Secondly, the I/O map lends itself to breaking the problem into subproblems that can be solved somewhat independently. In larger problems than used in this illustration, ones with multiple outputs (and thus ladder solutions with multiple rungs), each output has its own rung and the logic for each rung can often be determined independent of the rest of the system.

Finally, the intermediate steps mean that cognitively, the programmer can navigate to a solution in many small cognitive steps rather than a few large cognitive leaps. The smaller steps make the solution approach more psychologically realistic, and the problem solving comes more easily and quickly, with fewer mistakes.

4. Conclusion

Are all intermediate representations equally effective in bridging the gap between problem statement and standard formulation? The answer is emphatically no. Some textbooks on PLC programming, for example, recommend flowcharting as an intermediate step between problem and ladder program.¹² My experience is that flowcharting, while effective for many code-based programming languages, is NOT an effective intermediate representation for ladder diagrams. In fact, I've found that flowcharting actually makes PLC problems *more* difficult to solve in many cases. One can imagine trying to flowchart the fan control problem above, then deriving a ladder diagram from the flow chart—not a straightforward task. The reason for this is that the PLC microcomputer executes ladder programs in a simultaneous fashion. It scans all the inputs at one time, then updates all outputs at one time based on the ladder logic. A flowchart by its very nature is sequential. This disconnect can influence one's reasoning negatively because the flowchart can induce a reasoning process that is at odds with logical structure inherent in the solution formulation. Thus, creating a flow chart doesn't necessarily bring the programmer closer to a solution, and may in fact take him/her farther away!

The characteristics that constitute a “good” intermediate representation (IR) are the subject of further research. But cognitive theory and the PLC programming example point to a few initial indicators. 1) IR's should require activity on the part of the user. At the heart sociocultural theory is that human cognition stems from human activity, so the representation should get the user to do something. In the PLC example, it's the process of creating the I/O map that produces learning, not the end product. 2) The IR must have a consistency in logic structure with the solution formulation. Inconsistency produces internal mental processes that are at odds with the logic needed to solve the particular problem. The flowchart vs. I/O map is a case in point, with the latter being more consistent in structure than the former. 3) The intermediate representation should make key distinctions/insights visible. The goal of the IR is to help the user gain an in-depth understanding of the problem quickly. In the case of PLC's, the IR helps identify and classify the system elements, and enables grouping them into sub-problems amenable to solution. 4) This leads to a fourth characteristic of an effective IR: it enables breaking the problem approach into a series of sub-tasks. 5) Finally, the IR should be simple, flexible, and not prescriptive. It should not be a recipe that one follows blindly to get an answer—the whole point is to help the user think more clearly. Plus, it should be simple and flexible enough that it is useful for many different problems.

IR's already exist in many problem domains. But a question that should be asked is whether the current tools are the best possible. Free body diagrams seem to be a tried and true psychological tool. But in computer science, for example, some curricula teach psuedo-code as the main IR for planning computer programs to solve specific problems. I speculate that until one masters a computer language psuedo-code will be of limited value, and that perhaps a more suitable IR would help students right better code sooner. Additionally, IR's do not exist in many problem domains. It seems that research focused on developing new psychological tools aimed at problem

understanding and structuring would be highly beneficial.

Bibliography

1. Atman, C. J., J. R. Chimka, K. M. Bursic, and H. L. Nachtmann, "A Comparison of Freshman and Senior Engineering Design Processes," *Design Studies*, vol. 20, no. 1, 1999, pp. 131-152.
2. Waldron, M. B. and K. J. Waldron, "The Influence of the Designer's Expertise on the Design Process," in M. Waldron and K. Waldron (eds.), *Mechanical Design: Theory and Methodology*, Springer-Verlag New York, Inc., 1996, pp. 5-20.
3. Bodker, S., "Understanding Representation in Design," *Human Computer Interaction*, vol. 13, 1998, pp. 107-125.
4. Johnson, S., "What's in a Representation, Why do we Care, and What does it Mean? Examining Evidence from Psychology," *Automation in Construction*, vol. 8, 1998, pp.15-24.
5. Suchman, L. A., "Representing Practice in Cognitive Science," in M. Lynch & S. Woolgar, eds., *Representation in Scientific Practice*, MIT Press, Cambridge, MA, 1990.
6. Latour, B., "Drawing Things Together," in Lynch, M & Woolgar, S, eds., *Representation in Scientific Practice*, MIT Press, Cambridge, MA, 1990.
7. Zhang, J., "The Nature of External Representations in Problem Solving," *Cognitive Science*, vol. 21, no. 2, 1997, pp. 179-217.
8. Adelson, B., "Cognitive Research: Uncovering How Designers Design," *Research in Engineering Design*, Vol. 1, 1989, pp. 35-42.
9. Wertsch, J. V. and P. Tulviste, "L. S. Vygotsky and Contemporary Developmental Psychology," *Developmental Psychology*, vol. 28, no. 4, 1992, pp. 48-557.
10. John-Steiner, V. and H. Mahn, "Sociocultural Approaches to Learning and Development: A Vygotskian Framework," *Educational Psychologist*, vol. 31, no. 3, 1996, pp. 191-206.
11. Zhang, J. and D. A. Norman, "Representations in Distributed Cognitive Tasks," *Cognitive Science*, vol. 18, 1994, pp. 87-122.
12. Webb, J. W. and R. A. Reis, *Programmable Logic Controllers: Principles and Applications*, 4th edition, Prentice Hall, Upper Saddle River, NJ, 1999.
13. Sobek, II, D. K., "Teaching Novices How to Program PLC's," to appear in the Proceedings of the 2001 ASEE Conference, Albuquerque, NM, June 24-27, 2001.
14. Sobek, II, D. K., V. A. Cundy, and V. L. Briggeman, "Assessing the Given-Find-Solution Method in an Undergraduate Thermodynamics Course," *International Journal of Mechanical Engineering Education* (under review).

DURWARD K. SOBEK, II

Durward Sobek is currently Assistant Professor of Industrial and Management Engineering at Montana State University. He holds a B.A. degree in Engineering Science from Dartmouth College, and the M.S. and Ph.D. degrees in Industrial and Operations Engineering from The University of Michigan. His current research interests are in the areas of new product development and engineering design education. Please visit his web site for additional information: <http://www.coe.montana.edu/ie/faculty/sobek>.