Everyday Problem Solving in Engineering: Lessons for Engineering Educators

DAVID JONASSEN
Educational Psychology and Learning Technologies
University of Missouri

JOHANNES STROBEL
Educational Technology Programme
Concordia University

CHWEE BENG LEE
Learning Sciences and Technologies Academic Group
Nanyang Technological University

ABSTRACT

Practicing engineers are hired, retained, and rewarded for solving problems, so engineering students should learn how to solve workplace problems. Workplace engineering problems are substantively different from the kinds of problems that engineering students most often solve in the classroom; therefore, learning to solve classroom problems does not necessarily prepare engineering students to solve workplace problems. These qualitative studies of workplace engineering problems identify the attributes of workplace problems. Workplace problems are ill-structured and complex because they possess conflicting goals, multiple solution methods, non-engineering success standards, non-engineering constraints, unanticipated problems, distributed knowledge, collaborative activity systems, the importance of experience, and multiple forms of problem representation. Some implications for designing engineering curricula and experiences that better prepare students for solving workplace problems are considered.

Keywords: problem-solving, workplace, engineering practice

I. INTRODUCTION

For years, reports have validated the importance of problem solving in the workplace. For instance, the SCANS Report (The Secretary's Commission on Achieving Necessary Skills) [1], *What Work Requires of Schools*, states that problem solving is an essential skill for workers. ABET specifies the abilities to identify, formulate, and solve engineering problems as essential learning outcomes for any engineering program. "If the United States is to maintain its economic leadership and sustain its share of high-technology jobs, it must prepare the engineers of tomorrow for future technological and societal changes and to acquire new knowledge quickly and apply it to emerging problems," said Wayne Clough, Chair of the Committee on the Engineer of 2020 [2]. Engineers are hired, retained, and rewarded for their abilities to solve workplace problems.

If engineering education programs are to meet these challenges, they must comprehend the nature of workplace problem solving in order to better prepare their graduates for the workplace. In this paper, we support that understanding by describing qualitative research studies that identified the parameters of everyday problems solved by practicing engineers. These parameters can be used to contrast the kinds of problems most often solved in the workplace with the kinds of problems that are most often solved in engineering classrooms. From that perspective, this study is a needs assessment aimed at articulating the nature of the problems that engineering students must learn to solve if they are to be successful in the workplace (the educational goal state). The kind of problems most often encountered in engineering programs (except for capstone and assorted design experiences) is the story (word) problem, for which the parameters of a problem are specified in the problem statement. Word problems possess knowable, correct solutions that are achieved by applying preferred solution methods; and they apply a limited number of regular rules and principles that are organized in a predictive and prescriptive arrangement [3]. When learning to solve story problems in engineering, students learn to translate relationships about unknowns into equations, solve the equations to find the value of the unknowns, and check the values found to see if they satisfy the original problem [4]. This linear process implies that solving problems is a procedure to be memorized, practiced, and habituated, a process that emphasizes getting answers over making meaning [5].

In order to solve workplace problems, students must develop adequate conceptual frameworks (make meaning) and apply those frameworks in solving complex ill-structured problems. Ill-structured workplace problems have vaguely defined or unclear goals and unstated constraints; they possess multiple solutions and solution paths or no consensual agreement on the appropriate solution; they involve multiple criteria for evaluating solutions; they possess no explicit means for determining appropriate actions or relationships between concepts, rules, and principles that are used; and they require learners to make judgments and express personal opinions or beliefs about the problem and defend them [3]. Educators historically have assumed that learning to solve well-structured problems positively transfers to solving ill-structured workplace problems. However, recent research has shown that learning to solve well-structured problems does not readily transfer to ill-structured problems [6–8]. That is, learning to solve story problems in engineering classes does not enable graduates to solve complex and ill-structured workplace problems. Different kinds of problems engage and require different cognitive processes [9].

This paper describes the process that we used to identify attributes of workplace problems and to articulate the context, activities, and constraints that make workplace problems so ill-structured. The findings of this study may advise the design of authentic problem-solving experiences for engineering students.

II. Methodological Framework

As a methodological framework, we employed a modified analytic induction process, a qualitative research methodology that uses a systematic set of procedures to develop an inductively derived grounded theory [10]. Analytic induction provided an ideal methodology for identifying themes and categories within engineering stories told by practicing engineers. By utilizing a multi-case design, several engineering experiences were compared and contrasted. In the analytical induction approach, data built the basis for further descriptions and interpretations, but as the term induction indicates, the methodology did not employ an atheoretical empiricism, but rather was informed by prior research. Therefore, this research methodology is the most appropriate for answering the fundamental research question about the nature of workplace problem solving.

In order to systematically identify our assumptions and derive new aspects of problem solving informed by data, we utilized two approaches: (a) the development of a case-based reasoning (CBR) library indexing semi-structured interviews with engineers based on our assumptions and existing problem-solving research followed by (b) a grounded theory approach to elicit new perspectives on workplace problem solving that are informed by the CBR indexes but go beyond them.

III. Study 1: Case Library of Engineering Experiences

In order to orient its research agenda and to identify its own assumptions, the members of the NSF-funded Center for the Study of Problem Solving developed a case library of engineering problems. The case library consisted of transcribed stories of problem-solving experiences of practicing engineers. Case libraries are the product of applying case-based reasoning (CBR) processes. CBR is an artificial intelligence formalism for representing phenomenological (experiential) forms of human intelligence. According to CBR, stories constitute an important form of human intelligence. In any new problem situation, people examine the situation and attempt to retrieve previously encountered experiences that resemble the current one. Along with information about the situation, we retrieve the lessons that the situation taught us. Engineers, like most workers, solve problems by remembering similar cases and applying the lessons learned from those cases to the new one [11] because "they work with case histories and use narrative explanations to understand why the people they work with behave the way they do" (p. x). The CBR process is a cycle of activity in which a newly encountered problem (the new case) prompts the reasoner to retrieve cases from memory and to reuse the previous case (i.e. interpret the new in terms of the old) to solve the problem [12–14]. If the suggested solution will not work, then the previous case is adapted and tested. If the problem is solved, then the new experience is indexed and retained by the problem solver in his/her own case library. CBR is a process of externalizing previous experiences and indexing and storing representations of those experiences in a database for reuse by less experienced problem solvers.

A number of research studies have illustrated the importance of stories in workplace problem solving. Klein and Calderwood [15] found that experts (fire commanders, tank commanders, and system designers) relied more heavily on cases based on past experience than on abstract principles when making decisions in situations with a high degree of uncertainty. The stories they recalled focused on situational awareness and understanding expected outcomes. Ross [16-17] found that people learning a new skill naturally use what they have learned from solving a previous problem and apply it to the new problem. Lancaster and Kolodner [18] found that automobile mechanics frequently use their experiences and those of others when wrestling with new problems, while Kopeikina et al. [19] found similar evidence with GTE engineers who were troubleshooting phone-switching networks. The reuse of problems is essential to learning how to solve problems. Engineers naturally reuse their problem solving experiences or call on others to recount their experiences when solving problems. The purpose of this first study is to describe the dimensions of those problems.

A. Method

During the spring and summer of 2004, we conducted structured interviews with 106 volunteer practicing engineers solicited from the ranks of the Missouri Society of Professional Engineers. Based on a generic model of problem solving, we identified important indexes that we would use to build the case library of problem-solving stories. Those indexes and the results obtained from the study are presented in Table 1. The interview focused on a single job or project completed by each engineer at some point in his or her career. The engineers were asked to recall a typical problem they had solved in the past. We began each interview with questions regarding the engineers' background, the organizational context in which they worked and later asked questions about the nature of the problem, how they analyzed and represented the problem, how they generated solutions, and how successfully the job was completed. The interview questions appear in the Appendix. The questions were designed to elicit information required to construct the case library. Ninety-seven interviews were completed and transcribed (nine interviews were incomplete or the transcriptions were not able to be interpreted).

Next, we indexed the transcripts of the interviews. Indexing is the process of assigning labels to cases at the time that they are entered into the case library to ensure that they can be accessed and retrieved at appropriate times [20]. The index structure for the case library was derived from literature on problem solving and discussions with engineers. It was adapted several times based on early pilot interviews. The indices that were assigned to each project along with a summary of the results are shown in Table 1.

Excerpts from each interview representing each of the indexes were stored in an Oracle database searchable by a case-based reasoning engine (csps.missouri.edu). First, all cases indexed in the case library were converted to numeric values called case feature vectors according to the index structure listed in Table 1 and put in the high dimensional vector space. Second, a user querying the case library identifies the aspects of the engineering problem (context or situation) that are most relevant. Then, the user interface turns the problem into a query case, which in turn is converted into a query feature vector. Third, the query vector is matched against all case vectors in the high dimensional vector space using the nearest neighbor algorithm that calculates the weighted distances between the query vector and all case vectors. Fourth, the search engine returns all matched cases with case numbers and abstracts ranked in distance. A shorter distance means a closer match. Then, the user can read abstracts of the retrieved cases and choose a case number to read the complete transcript.

B. Results

While the purpose of constructing a case library of engineering stories was to provide a pedagogical support system and a knowledge base for further analysis, the results provide important descriptive information about the nature of everyday problem solving in engineering as well as feedback about the accuracy and assumptions of our indexes. For each index in the database, the percentage of the cases that were assigned a particular index (unless otherwise stated) is listed in Table 1.

Based on the indexing process, we recognize that the sample is biased in the direction of civil engineers and civil engineering projects because of their disproportionate representation in the sample. This imbalance was reflected in the membership of the Missouri Society of Professional Engineers. Most of the sample possessed a bachelor's degree in engineering working for large companies solving fairly large problems. Of most interest is the nature of the problems solved. The most common output is a design, which is generally regarded as the most common output of engineers. The large number of construction projects is a function of the disproportionate number of civil engineers in the sample population. As expected, these engineers worked with a disparate combination of other professionals and paraprofessionals. These engineers relied extensively on their own experience when solving problems. They used a variety of problem representation tools; however, they relied on brainstorming as much as decision analysis methods for understanding the problem and for generating solutions. As will be shown in the next study, unanticipated problems occurred in most projects and a variety of non-engineering constraints and solution metrics modified problem solutions. These findings set the stage for the second study, where we analyzed the interview protocols in more depth.

Engineering education: civil (39), electrical (18), chemical (10), mechanical (13), structural (5), industrial, nuclear (1), other (16)

Engineering education level: bachelors (70), masters (30), doctoral (0)

Professional engineering experience (3 years to 41 years, M=15.3 years)

Business size: 1-20 (21), 21-100 (29), 101-500 (17), 501-1000 (33)

Department size: 1-10 (40), 11-50 (22), 51-75 (7), 76-100 (31)

Problem size: large (45), midsize (43), small (12)

Department type: chemical (0), civil (14), electrical (20), industrial (0), mechanical (7), product development (11), safety (7), quality control (11, executive level (20)

Problem description: manufacturing (5), design (40), construction (36), supply chain (7), consulting (12)

Problem initiation: client (36), interviewee (24), boss (8), other (32)

Analyses used: visual inspection (21), data collection (31), calculation (32), modeling (5), interview (11)

Final Product: design (62), detailed plan (19), prototype (1), standards (3), recommendation (32)

Problem Representation: spreadsheet (14), 3D (12), mechanical drawing (15), formulae (8), other (40)

Fields of people involved (raw numbers): surveyors (10), designers (20), politicians (8), citizens (6), engineers (52), equipment vendors (6), technicians (26), other (40)

Experience with similar problems: frequently (50), occasionally (22), rarely (28)

Constraints (raw numbers): budget restrictions (33), time available (35), geographic constraints (12), zoning (5), other (55)

Problem completion time (weeks): 1-30 (29), 31-54 (17), 55-110 (20), 111-300 (34)

Solution metrics (raw numbers): dollar (22), size (1), customer satisfaction (43), completion (27), other (35)

Knowledge sources (raw numbers): experience (50), manuals (18), client (9), other people (32), other (19)

Unanticipated problems occurred: yes (90); no (10)

Solution generation methods: brainstorming (33), decision analysis (37), company procedure (16), other (14)

Table 1. Indexes and values used to describe each project and engineer, stated in parentheses as percentages (unless otherwise noted).

IV. Study 2: Qualitative Analysis of Interviews

CBR indexing began with a conceptual model of the stories based on a rational analysis of the problem-solving process. In order to provide a different perspective on the stories that engineers told, we decided to qualitatively analyze these interviews, making fewer assumptions about the outcomes. We treated the interviews as multiple case studies, on which a grounded theory approach was employed.

Coding the interview transcripts began by examining the interview protocols for salient categories of information that were supported by the text and then identifying categories of information. We used the qualitative research tool, Qualrus, to record themes and categories and associate interview text with those categories. Ten cases were coded by three different raters in order to validate the emergent codes. Thereafter, all cases were coded by a single rater, whose work was checked by the other two coders. A total of 78 cases were coded. The categories or themes either emerged from the data or were informed by the results of the case library. The coding of the categories was the means for establishing similarities and differences among the cases and for redefining the categories/ themes during the process. After the initial process of open coding, axial coding reduced the first set of codes by merging similar codes and reassigning codes to particular text elements. The following primary themes emerged during qualitative analysis and were agreed upon by three raters.

A. Theme 1. Workplace Problems Are Ill-Structured

A recurring characteristic of workplace problems is that they are ill-structured. Initially, some problems appeared fairly well-structured, however, as constraints and unanticipated problems (described later) became apparent, the problems became more ill-structured. For example, one problem looked very well-structured: measuring the flow of a certain pipe and the temperature coming out of a large container. However, the engineer realized that in order to install a thermostat, the tank needed to be sealed and was then impossible to access. While the engineering activities often formed the core of the problem solution process, the entailments from those activities (working with other people, dealing with environmental constraints, and managing the project) made the problems more ill-structured. In another example, a relatively new mechanical engineer working on a design problem recounted:

> Probably the biggest challenge that we see in some of these projects is dealing with incomplete information. Invariably people won't know what the output is going to be for the product. So you don't know how to do piping designs, or they don't know what kind of cooling load or heating load [is] to be expected. You don't know the specific heat is because its not listed. Or any number of design parameters that are not defined. In some cases you are making assumptions in design and you're making critical assumptions that you can do what you're wanting to do based on some piece of information or lack of information.

Unanticipated problems are among the most common reason for problems becoming ill-structured, as recounted by an engineering manager on a construction design project:

> There were a lot of design-related challenges. I'll try not to go into too much detail other than to say that we had things such as there was so much reinforcing steel in the concrete monolith that the contractor couldn't use the size aggregate we'd specified because he couldn't get it through the reinforcing steel. So we had to come up with either changing the configuration of the reinforcing steel or changing the size of the aggregate. That took some testing before we could make a decision. That seemed to be the logical way to go, but we also looked at changing the rebar. It turned out that was not a good solution because it would take almost a total redesign of the monoliths.

Often, problems in projects do not occur until after the project has ostensibly been completed. This raises the issue of when a problem is solved. What obligations do engineers have to clients after they are paid for a project? A civil engineer provided the following example of problems that may develop after the original problem was solved.

> The problem just showed up 10 or 12 years after construction, and we never did really find out why it showed up 10 or 12 years after construction, but we did get the mix optimized and everything for that, and hopefully we won't have that problem in the future, but we're not sure exactly why, what was causing some cracking problems later on in life.

B. Theme 2. Ill-Structured Problems Include Aggregates of Well-Structured Problems

Within large projects, numerous well-structured problems are solved, such as "what is the load strength for material x" and how big is the radius of machinery to clear a path. Engineering students learn to solve well-structured problems in university courses; however, they rarely experience well-structured problems in everyday contexts. Two engineers provide relevant examples of this theme.

> We had to decide how big [to build] a lagoon to hold the dirt and the possible rain for the possible amount of time we were treating this soil. And we had to decide how best to treat the soil. We had to make calculations how long [it] would take [to] put back into the ground. As we are doing all of this [we had to] decide where we would sample the soil and separate [it]. We had to figure out what we were going to find in the hole, how we would treat it so what we would know about the size, and what to put in our water treatment system, and how we were going to power it. So those were a few of the decisions we needed to make.

> With all the data that we collected out in the field on the performance of them in the past years, we looked at which had the fewest cracks, which had come loose the most, which had the fewest repairs, and which were more impermeable to chlorite, to salt that gets down in them and corrodes the rebar. It was analyzing the data and then also trying to confirm that with some other state DOTs that had used the same thing.

C. Theme 3. Ill–Structured Problems Have Multiple, Often Conflicting Goals

The ability to solve any kind of problem is an understanding of what kind of problem it is, that is, what the goal of the problem is.

In textbook problems, the goal is obvious, but in workplace problem solving, there are often multiple sub-goals that must be considered and reconciled to the main goal. For example, the primary goal of one project was to "find a solution that will meet the purpose and needs statement that we include in our environmental impact statement that has a level of public and community support along the corridor that is politically acceptable and ultimately that we can afford." Accommodating the goals and expectations of each of those factors is a complex undertaking. Sub-goals can often conflict with the primary goal, so the engineer must determine which goals have higher priority. Often those goals have nothing to do with engineering outcomes. Two experienced civil engineers recall exemplary situations from large construction projects.

> We'll measure it with a variety of things. Number 1, did we meet the anticipated goals for hiring of a diverse work group. That is part of the contractual requirements as well as the participation of a variety of different kinds of enterprises. Our safety record, and of course, did we make any money on the project?

> Our job was to actually do the work. To operate the machines that cleans it, to build the lagoon and volatize it and to build water treatment systems to treat the water.

Even the bottom line contains multiple goals, as expressed by a construction engineer.

> Our goal is to always work safely and make money. Safely and on time and make our client happy and to get additional work from our client.

D. Theme 4: Ill -Structured Problems Are Solved In Many Different Ways

In textbook problems, there is a preferred solution path or method. However, workplace problems are ill-structured because they have multiple solution paths, that is, there are a variety of methods that may be used to solve the problem. In many ill-structured problems, the problem solvers never know which solution method is optimal or even how to evaluate the efficacy of different solutions. They use their professional judgment or rely on their experience (as described later). The following excerpts illustrate this theme.

> But we looked at atleast 3 different ways of doing it, and there are a whole lot more ways.

> There are probably several different ways to solve it technically.

> There were a number of possible solutions.

> Yes. Usually there are always several ways to resolve a problem.

> Typically, when a structure is over a half-million dollars, we'll experiment with various sized models and other alternatives to try and reduce that cost.

> We usually consider dozens.

> You keep playing with different formulations, different ways of building a tire, to get less rolling friction. We do lots of different things.

> A project is comprised of multiple decision analysis paths.

The implications of this theme for engineering education are obvious. Rather than engaging students in problem solving where the correct solution method is obvious, students must learn to identify and evaluate multiple solution methods.

E. Theme 5. Success Is Rarely Measured by Engineering Standards

Engineering classroom problems often assume that engineering problems are solved using only engineering criteria as the criteria of success. Although solutions to workplace problems must meet implied or explicit engineering standards, according to our data, those are rarely the standards that are used to describe the success of a project. From an engineering perspective, the ultimate engineering criterion is failure [21]. Virtually every calculation that an engineer makes is a failure calculation. However, the success of engineering projects is rarely measured by engineering standards alone. For most engineers, the most common criteria they are held to are satisfying the client, completing the project on time, and staying under budget (e.g., "did we make any money on the project?" or "is the client is happy?"). Even when asked about different solutions and how they interplay, rarely were more sound engineering solutions mentioned. In order to please the client and make money, numerous other criteria are often applied. For example, legal, regulatory, and environmental criteria become the arbiters of success.

> So we are pretty savvy as to understanding what the code is trying to say. You have some people in these code making bodies that you can consult to make formal interpretations and written interpretations. So we had to make sure the bank would give us a line of credit and we had to talk to our client to see if they would pay us some up front money to start building these systems. Funding was a big concern, and we have to make sure we have legal constraints as long as we are complying to the law, and we want to make sure the client is not asking us to do something illegal. We also want to make sure that we have a contract where every party is happy, if that is possible. Make sure we get paid, and they understand what we are going to do and we understand what their expectations are.

F. Theme 6. Most Constraints are Non-Engineering

Most engineering education programs treat problems as engineering-only problems. However, workplace problem constraints, like standards, usually had little to do with engineering. Rather they most often related to time ("We had a very aggressive schedule from start to finish") and budgets ("Dollars, it is always dollars."). When the clients are other companies, the constraints are determined by cost, functionality, and the requirements that new solutions have to work together with elements already in place, such as overall corporate brand, jobs, tasks, and tools that are already in place. Workplace problems were more complex and ill-structured because of political constraints, such as regulations or acceptability to citizens; environmental constraints, such as requirements to meet environmental regulations, or obtain permits; economic constraints, most often dealing with the budget; and cultural constraints, such as the corporate culture or local context. Several engineers described constraints that were based on biases, such as the following:

> We are solving a whole series of things in the fact that an architect or owner wants to build a building a certain way, and he has certain needs and desires, but yet we have all these safety codes that need to be met.…the state … building and Fire Department that had their whole set of requirements. And we had to make sure that those requirements also didn't pose problems. So we had a whole series of requirements one playing off against the other that we had to balance out. And there were several environmental issues up there, concerns from U.S. Fish and Wildlife that we were going to hurt the fish.

Those biases were often personal preferences on the part of the client and just as often constrained by codes or other legal restrictions as well as financial constraints. Problems were frequently subject to multiple non-engineering constraints.

Constraints are often based on communication problems, such as:

> Accessibility of people that needed to give us input. We did in this substation where this cable is coming out, and they are doing some other work in there. Another engineering firm is going to do some work for them. And it is hard for us to coordinate with them because they haven't been brought on board yet because they don't have their budget approved. That has been an obstacle too.

Our preliminary research on design problem solving has shown that constraint analysis is an important part of the engineering design process. One engineer supported our research by claiming, "You have to put the constraints into the beginning of the process so that they're an identical part of the process."

G. Theme 7. Problem Solving Knowledge Is Distributed Among Team Members

Traditional conceptions of learning have focused on knowledge that is acquired by individuals. Early theories of cognition focused on information processing and knowledge in the heads of individuals. According to newer perspectives, learning is less a solitary act of individuals but rather is distributed among people, their tools and communication media, history, and the artifacts they create. Knowledge exists not only in the heads of learners, but also in the conversations and social relations among collaborators [22].

Hutchins [23] provided one of the most eloquent accounts of distributed cognitions, or as he referred to it, "cognition in the wild," by describing how navigating a ship through a narrow passage into port requires the coordination of multiple people and devices. People use instruments to take bearings and depth measures. One person records all of this information in a log, and another uses plotting instruments to compute the ship's position and course. Timing and coordination of these activities are critical. The navigator recommends changes to headings and speed to the deck officer, who may or may not accept them. The deck officer passes them onto helmsman who steers ship.

Why is a distributed perspective on cognition appropriate for analyzing engineering problem solving? Because engineers rarely work alone, they rely on the knowledge of many people to solve workplace problems. They work in "distributed networks of expertise," [24] where different team members contribute their skills and knowledge to the solutions of engineering problems. Engineering knowledge required to solve problems is usually distributed among a variety of people, including draftspersons, surveyors, other engineers, and administrators. Because of the size of the companies and problems described by the engineers in this study, the amount and diversity of knowledge required to complete the projects was high. Even in small companies, however, engineers nearly always rely on others' knowledge in order to solve problems. An experienced mechanical engineer recalled:

> There are 30 people in my group, and they are all working on various aspects of problems probably about half of them are software engineers and they are working on issues relative to that, and the rest of them or hardware or mechanical engineers.

Civil engineering projects, because of their size, are almost always very distributed.

> Typically we have a team of engineers on any project. It would typically be design engineers, mechanical mostly and electrical. And on each team we have HUE and a purchasing person or advanced manufacturing and a manufacturing rep as well. Pretty much a fully cross-functional team including a marketing guy.

> I oversee several sections, I oversee our design section, our planning section, our right away section and we have a general services section. I do that too. At the owner's facility, he had probably six or eight engineers involved. We had three or four operations people. And then we had two or three maintenance people involved. And then plus we had contractors. We had mechanical contractors doing the ductwork, installation, we had electrical contractors doing the wiring. And then we had a sub-contractor at Honeywell, who provided modifications to their control system.

In addition to distributing responsibilities among members of the same organization, most engineering problems require institutional knowledge found in several organizations, regulatory bodies, and support systems.

> There's certainly the property owner, there's the telephone company regional manager, there's the utility company chief engineer, there's the utility company distribution engineer, there's the utility company attorney. There's the utility company's insurance company attorney. There's the homeowners, the homeowners' insurance company attorney. There's his insurance adjuster. There's a fire investigator, two fire investigators, that's about all, oh there's my electrical testing company technicians.

> Inside our organization engineers, partners, cad drafters, graphical, computer, secretarial help. Outside of our firm we interfaced with the architect, the owner, the structural engineer, mechanical engineer, electrical engineer. We

> interfaced with all those disciplines because it is essential to have all those things working together as a package.

Cognitive abilities may also be distributed across time and minds. Pea [25] claimed that even intelligence is distributed; it is "manifest in activity that connects means and ends" (p. 50). For example, engineers appear intelligent because of their ability to use calculus to represent complex problems. Were it not for Leibnitz and other mathematicians who conceptualized differential calculus, engineers would not appear as intelligent, so their intelligence is historically distributed.

The ability to solve workplace problems is distributed throughout different activity systems. Individual engineers are not required to solve workplace problems independently. Individual students cannot learn to fulfill all problem-solving roles. Rather, students must learn how to interact with different people and systems and learn to rely on their advice and knowledge. Rather than being assessed exclusively for the knowledge that resides in their heads, they should be assessed for their abilities to use their own skills and abilities when appropriate and to call on others' expertise when appropriate.

H. Theme 8. Most Problems Require Extensive Collaboration

Very few engineers engage in solitary problem solving. In the overwhelming majority of workplace problems, engineers must collaborate with a variety of personnel in order to identify and solve the problem. Collaborations are most successful when the roles and relationships are well defined, and (like any good system), they share a common goal.

> We all pretty much know our roles but know that in our specialization those people touch on certain things affect the fire protection engineering and life safety.

> We are all working together for a common goal, which is to make sure that we have an economically viable building and a safe building-one that is going to function the proper way. We all sit down at the conference table together and we come up with a plan and then we work very closely with the engineering disciplines so we have all the details ironed out.

In considering the implications of this theme, we must better understand the intersections between separate teams of engineers. How do they interact? How much knowledge about the other person's work is necessary to get one engineer's work done? Follow-up studies should more closely examine the nature of these interactions.

I. Theme 9. Engineers Primarily Rely On Experiential Knowledge

Research has confirmed that experience is the most common determinant of expertise, and that the recall of historical information is the most frequent strategy for solving problems [26]. For example, Bereiter and Miller [27] found that troubleshooters base their diagnosis on their beliefs about the cause once a discrepant symptom is found. Those beliefs are based on historical information (i.e., experience). They also found that the most common reason for taking a particular action during troubleshooting is to test for the most common problem based on experience. Learning to solve problems begins in school with the construction of a conceptual model of the domain. After school, as the problem solver gains experience, their conceptual knowledge becomes embedded in their experiences [28]. They come to rely more on their historical knowledge of problems they have solved than their conceptual understanding.

Problem solutions to workplace engineering problems are based more on experience than engineering knowledge according to our interviews. Experience recommends solutions. These brief responses from engineers illustrate how essential prior experience is when solving problems.

> That was just based on experience with our systems.

> You build on your previous experience.

> Experience some problems like [those] that have occurred in the past. Experience on those things is probably the biggest way that we get them solved quickly anyway.

> We would pull upon our past experience as to, maybe we run into similar cases before that would give us some insight on how to solve this problem.

> Most of the time these are experience projects.

> We have the assets (mostly human assets) with the right kind of experience and expertise to pull this project off.

Perhaps more importantly, experience helps to prevent failures.

> There are typical things that go wrong. A lot of it is just experience. We have been involved with this a long time and a lot of times we can tell by looking at what the problem is and by asking questions of the inspectors and production people. You usually get to the root of it pretty quickly.

> Some they agreed to and some they felt like their experience indicated that it wouldn't work as good in this situation.

Most engineers feel that experience is required to convert the theoretical lessons from the classroom into problem-solving abilities.

> Through experience you obtain the judgment to apply that education base in the appropriate level of detail and analysis to the particular problem at hand.

> In a formal classroom setting most of the schools teach you the basics and principles of engineering. And maybe those principles don't include the physical tools but then when you get out there and understand the principles of engineering and apply those principles through problem solving you know what tools to pick up.

J. Theme 10: Engineering Problems Often Encounter Unanticipated Problems

Most everyday problems are dynamic; that is, the conditions change over time. Most of the problems the engineers talked about were large scale, in which a set of problems (some of which were unanticipated) occurred. What is interesting is that the unanticipated problems that arose were a combination of engineering and non-engineering problems, as described by an electrical engineer.

> Also at different times we don't live in a perfect world and when buildings get put together at times people can make mistakes. Sometimes they can't be rectified and need to be ripped out and other times where it would be disastrous to do that so we develop equivalency concepts for that. The other unanticipated thing is you can get in a project and the owner can change his mind and all of a sudden the whole dynamics of the project changes.

Often, unanticipated problems can aggregate, making the projects even more complex to solve.

> We had some unusual rain and we planned on operating the water treatment about eight hours a day while we were down there working occasionally. Because of the rain we had to send men down there over night or over the weekend to operate the water treatment system. As we dug they didn't tell us what used to be there. We found pipes with asbestos on it. We found some unexploded ordinance. This used to be an Army base where they used to make rockets, so the client didn't provide us with good information as to what we might find.

> The contract wasn't specific enough. We were working for another consultant. The chain of command was the client's consultant hired our consultant who hired us. We could only talk to our consultant and some of these things have to be made right then. And the chain of command delayed things a lot. If we had it do over we would have made it easier to get an answer more quickly.

> We encountered some chemicals in our water treatment that I did not expect. We were able to adjust, but it took a little while. If we had it to do over, we would have done our own sampling to confirm what was in the water. We did not get a lot of things in writing and later we had trouble collecting. Maybe writing up change orders would have been good.

This latter problem emphasized the need for clear communications among engineers, a theme that is reflected in recommendations from the engineers (12th theme). One engineer summarized this theme of unanticipated problems quite well:

> For a project that size, it's my experience that you're always going to have things come up that you don't anticipate.

K. Theme 11. Engineers Use Multiple Forms of Problem Representation

Representing the problem space is an essential part of all problem solving [29]. Experts are able to represent problems in multiple ways, whereas novices are typically restricted to a single form of problem representations [30]. Representing problems to others directs further interpretation of information about the problem, simulates the behavior of the system based on knowledge about the properties of the system, and triggers particular solution schemas [31]. Externalizing mental representations determines what information can be perceived, what processes can be activated, and what structures can be discovered from the representation [32]. So building models of problems using a variety of tools supports solutions of complex workplace problems.

Engineers use a variety of problem representation methods. The most common form of problem representation is a drawing, however "if it were a big job, we probably would make a chart." Drawings are most often rendered by hand or by using CAD software. However, a variety of other tools may also be used.

> We use a variety of tools depending what the project is and what the size and its complexity. Very often if the job is a steel frame we use a program called RAM structural system. My understanding developed for steel frame and moved on into accommodating other materials. We also use just a generic 2D frame analysis"

> We did mathematical modeling of the key elements of our process that controlled that quality characteristic. We performed validation experiments of that mathematical model.

> So is it going to be an actual model or a computer model? We will do both. We will model it in the computer first and then build hardware.

> We just use EXCEL and make it into a spreadsheet.

> We used a software package called Criteria Decision Plus that is a very analytical analysis of alternative[s] and decisions. You apply weightings to your various selection criteria and then you grade each alternative against those criteria.

The implication of this theme is also evident. Engineering students should not rely exclusively on algebra, calculus, and trigonometric formulas to represent problems. Our research confirmed that a small minority of workplace engineers regularly use mathematical formulas to represent problems. We are not suggesting that mathematical formulas should not be used in college classrooms, but rather that students supplement them with alternative, qualitative problem representations.

L. Theme 12. Engineers Recommend More Communication Skills In Engineering Curricula

Communication is the "sine qua non" of cognition [23]. Individuals may have mental representations derived from experience or observations, but that knowledge is often useless unless it is shared. Components of distributed systems, such as workplace problem-solving teams, must trust and rely on each other; no one person is in possession of all the information needed to make a decision.

Most engineers felt well prepared for core engineering jobs, however, there was general acceptance among most engineers that graduates will "really" learn how to be an engineer during the first year or two on the job. Rarely did practicing engineers recommend more engineering in the engineering curricula. Rather, most of the engineers emphasized more instruction on client interaction, collaboration, making oral presentations, and writing, as well as the ability to deal with ambiguity and complexity. As two engineers opined:

> It is kind of a sore spot with me that educational institutions teach when you do your work there is a right answer and a

wrong answer. And in the real world it is never that way, there are many ways to do things and it is not a matter of getting a right answer it's a matter of working for the best solution for your particular situation.

In school you have to do your own work and you're expected not to cheat and in the business world you solve everything on a cooperative basis. Make sure engineering-wise, in addition to their raw engineering they have good written and communication skills and make sure that they don't get tunnel vision.

Although workplace problems may not resemble classroom problems, academics still have their place in the workplace problem solving process.

Therein lies the big rub. You get the academics, people who have spent their entire career at universities, getting involved in design, and a lot of times they've never dealt with the reality. And we always have academics in our design team because these are the guys who know all the theories and you just flat want them out there. So it's a big combined effort of several different disciplines.

V. Implications for Engineering Educators

This study was intended to explicate some of the parameters of everyday, workplace engineering problem solving. Those parameters may be used by engineering education programs to design learning experiences to better prepare students to meet the challenges of ABET and the *Engineer of 2020*. The question that is obviously begged by this study is how to better prepare engineering students for solving workplace problems. In posing this question, we recognize the numerous, successful efforts to achieve this goal that have been made by many engineering programs. We offer the results of this study as guidelines for revising the nature of problem-solving instruction in engineering programs.

A. Workplace Transfer

An underlying assumption of this study is that a significant, if not exclusive, goal of engineering programs should be to foster workplace transfer. The traditional concept of transfer describes the ability to generalize solution methods from one problem (typically a decontextualized word problem) to another, similar word problem embedded in a different context. Bransford and Schwartz [33] expanded the concept of transfer to accommodate preparation for future learning, that is, acquiring the learning skills that will be required of learners in future learning situations, in school or out. For engineering programs, preparation for future learning in work situations should be the goal, acknowledging the all-too-common belief that learning ceases at graduation. In modern engineering contexts, the need for continuous, lifelong learning has never been greater. Therefore, for professional engineering programs, the clearest purpose for learning is preparation for future work, which includes the ability to solve problems and to learn independently and collaboratively. Because solving well-structured problems in science and engineering classrooms does not readily lead to solving complex, ill-structured workplace problems [6–8], engineering programs must support learning to solve complex, ill-structured workplace problems if they are to prepare their graduates for future learning and work.

B. Problem-Based Learning

One solution for preparing engineering graduates to become better workplace problem solvers is converting their curricula to problem-based learning (PBL). PBL programs replace traditional courses with integrated, interdisciplinary sets of complex problems that students learn to collaboratively solve. Student learning is self-monitored and self-directed; students must decide what knowledge they need to construct in order to solve the problems. Several engineering programs around the world (e.g., Aalborg University on Denmark, McMasters University in Canada, Monash University in Australia, Manchester University in England, Glasgow University in Scotland, Eindhoven University in the Netherlands, and Republic Polytechnic in Singapore) deliver the majority of their engineering curricula via PBL. Additionally, PBL modules or courses have been implemented in numerous engineering programs, including biomedical engineering [33], chemical engineering [34], software engineering [35-36], thermal physics [37], design processes [38], aerospace engineering [39], computing [40], microelectronics [41], construction engineering [42], and control theory [43]. Conversion to PBL requires systemic reform of curricula or at least entire courses. Although they have proven incredibly successful in many contexts, the level of commitment to such an innovation is more than most programs or professors are willing to make. Even if such a commitment is made, PBL programs face the continuous challenge of populating their problem base with authentic problems that are informed by everyday practice. In order to do so, PBL programs need to establish and apply a systematic process of identifying attributes of workplace problems and respond to critical changes in these problems over time.

C. Complex, Ill-Structured Problems

Although PBL represents one of the most important pedagogical innovations in the history of education (we believe), most classroom and many PBL experiences do not adequately accommodate the nature of workplace problems in their learning experiences. If the goal of engineering education programs is to better prepare engineers for the workplace, more classroom experiences and all PBL programs should engage students in resolving the complexities and ambiguities of workplace problems more consistently throughout the curriculum (not just in capstone contexts). That is, at least some of the problems that students learn to solve in engineering classrooms should require them to:

- Analyze and solve combinations of well-structured problems
- Manage multiple sub-problems
- Deconstruct multiple, often conflicting goals from a problem statement and analysis
- Reconcile multiple, conflicting constraints and criteria
- Analyze and select from a variety of solutions to various problems and to justify their selected solutions
- Identify and reconcile methods for achieving non-engineering criteria for solving problems
- Communicate and collaborate with a variety of professional and paraprofessional team members on all aspects of the

problem-solving process

- Anticipate and reconcile intervening problems and perturbations to the problem-solving process
- Adapt to changing project conditions and unanticipated problems
- Use multiple tools and formalisms (visual, verbal, quantitative) to represent problems
- Experience directly or vicariously the complexities of workplace problems as often as possible ("they should have some classes or something where you could have got to go out in the field a little bit and see some of this stuff").

Engineering programs have for many years provided internship experiences to students that are intended to engage engineering students in these kinds of experiences. Those experiences are generally deemed invaluable to the intellectual and professional development of engineering students. However, internship experiences are subject to the limitations of all apprenticeship experiences. For safety or productivity reasons, apprentices are often relegated to non-essential, inauthentic tasks. They rarely have the opportunity to encounter a substantial range of engineering problems or take risks that are an inherent part of real problem solving. In order to assess the quality of internship experiences, the attributes of workplace problems that we have articulated may be used as criteria.

D. Different Kinds of Problems

Another implication of this study is to engage engineering students in solving as many different kinds of problems as possible. Many engineering programs are incorporating design experiences throughout their curricula. Design problems are the most complex and ill-structured of all kinds of problems [9], and there are different kinds of design problems [45]. Despite the apparent goal of finding an optimal solution within determined constraints, design problems usually have vaguely defined or unclear goals with unstated constraints; they possess multiple solutions with multiple solution paths; and they possess multiple or unknown criteria for evaluating solutions.

Although design problems are the most common kind of problem that practicing engineers solve (designing products, processes, systems, and methods), engineers also solve a variety of decision-making, troubleshooting, and systems analysis problems, each of which calls on a different set of cognitive skills [9]. For example, when comparing the understandings of design engineers and maintenance technicians, designers' understanding emphasizes theoretical concepts whereas technicians emphasize experience [46]. When attempting to troubleshoot a system, designers required longer because they were sidetracked by what they perceived as design flaws [47]. In order to provide the most comprehensive preparation for workplace experience, engineering programs should also engage students in solving those kinds of problems as well.

E. Problem-Based Learning Environments

For engineering faculty who are committed to problem solving but do not have the support to develop PBL programs, they can with minimal support design, develop, and implement problem-based learning environments. These online environments provide problems to students in narrative form, representations of related problem-solving experiences, related cases, information resources required to generate solutions, cognitive tools for representing problem elements, and communication tools for supporting collaboration [48]. We are working on design architectures for scaling the development of story problems [49] and troubleshooting problems [50]. We describe an example of a problem-based learning environment to introduce undergraduates to the range of nuclear applications [51].

F. More Meaningful Collaboration

In order to address ABET requirements that engineers become able to function on multi-disciplinary teams, team work, and collaborative learning have become staples of engineering classrooms [52]. That need is supported by this study. However, too often, teams are formed on the basis of convenience rather than the skill sets of the participants or the roles that they play. It is also important to avoid bias or marginalization that underrepresented students often experience when participating in team related activities. Team related activities should be evaluated for the extent to which they engage meaningful communication that engenders a sense of ownership among a variety of stakeholders. They should also be evaluated for the meaningfulness of the collaboration, that is, are the roles that team members play diverse and authentic. Do the activities foster positive interdependence, individual accountability, promote interaction, social skills, and co-construction of knowledge [53] by engaging in authentic, collaborative tasks?

VI. Conclusion

This qualitative study identified the attributes of workplace engineering problems that make them complex, ambiguous, and ill-structured. Workplace problems often have conflicting goals, multiple solution methods, non-engineering success standards, non-engineering constraints, unanticipated problems, distributed knowledge, and collaborative activity that rely on multiple forms of problem representation. We discussed some implications for engineering education, including reconceptualizing the concept of transfer, solving different kinds of problems as well as problem-based learning and problem-based learning environments. The problems that engineering students do solve should exhibit some of the attributes that we identified in the study. Evaluating and extending these implications for engineering programs must precipitate research among the engineering education community to validate the most effective methods, and it should be dialectic, including all the important stakeholders in the conversation.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 0350305. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] U.S. Department of Labor, *What Work Requires of Schools,* Washington, DC: U.S Department of Labor, Secretary's Commission on Achieving Necessary Skills, (1991).

[2] Clough, W., *The Engineer of 2020: Visions of Engineering in the New Century,* National Academy of Engineering: Washington, DC, 2004.

[3] Jonassen, D.H., "Instructional Design Model For Well-Structured and Ill-Structured Problem-Solving Learning Outcomes," *Educational Technology: Research and Development 45* (1), 1997, pp. 65–95.

[4] Rich, B., *Schaum's Principles of and Problems of Elementary Algebra*, Schaum's: New York, (1960).

[5] Wilson, J.W., M.L. Fernandez, and N.D. Hadaway, *Mathematical Problem Solving*. Retrieved 9/22/05 from http://jwilson.coe.uga.

[6] Cho, K.L., and D.H. Jonassen, "The Effects of Argumentation Scaffolds On Argumentation and Problem Solving," *Educational Technology: Research & Development, 50,* (3), 2002 pp. 5-22.

[7] Dunkle, M.E., G. Schraw, and L.D. Bendixen, *Cognitive Processes in Well-Defined and Ill-Defined Problem Solving,* Paper presented at the Annual Meeting of the American Educational Research Association, San Francisco, California, April, 1995.

[8] Hong, N.S., D.H. Jonassen, and S. McGee, "Predictors of Well-Structured and Ill-Structured Problem Solving in An Astronomy Simulation," *Journal of Research in Science Teaching, 40* (1), 2003, pp. 6–33.

[9] Jonassen, D.H., "Toward A Design Theory Of Problem Solving," *Educational Technology: Research & Development, 48* (4), 2000, pp. 63–85.

[10] Bogdan, R.C., and S.K. Biklen, Qualitative Research For Education: An Introduction To Theory and Methods, Boston, Massachusetts: Allyn & Bacon, 1992.

[11] Polkinghorne, D., *Narrative Knowing And The Human Sciences*, Albany New York: State University of New York Press, 1988.

[12] Lancaster, J.S., and J.L. Kolodner., "Problem Solving in a Natural Task as a Function of Experience," *Proceedings, Ninth Annual Conference of the Cognitive Science Society*, Hillsdale, New Jersey: Lawrence Erlbaum Associates.

[13] Kolodner, J., "An Introduction To Case-Based Reasoning," *Artificial Intelligence Review*, 6 (1), 1992, pp. 3–34.

[14] Aamodt, A., and E. Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches," *Artificial Intelligence Communications*, 7 (1), 1996.

[15] Klein, G. A., and R. Calderwood, "How Do People Use Analogs to Make Decisions?," *Proceedings, Workshop on Case-Based Reasoning (DARPA)*, J. Kolodner ed., San Mateo, California: Morgan Kaufmann, 1988.

[16] Ross, B.H., "Remindings in Learning: Objects and Tools," in S. Vosniadou and A. Ortony, eds., *Similarity, Analogy, and Thought*, New York: Cambridge University Press, 1986.

[17] Ross, B H., "Some Psychological Results on Case-Based Reasoning," in K.J. Hammond, ed., *Proceedings: Second Workshop on Case-Based Reasoning (DARPA),* San Mateo, California: Morgan Kaufmann, 1989.

[18] Lancaster, J.S., and J.L. Kolodner, "Problem Solving in a Natural Task as a Function of Experience," *Proceedings, Ninth Annual Conference of the Cognitive Science Society*, Hillsdale, New Jersey: Lawrence Erlbaum Associates.

[19] Kopeikina, L., R. Brandau, and A. Lemmon, "Case-Based Reasoning for Continuous Control," in J. Kolodner, ed.., *Proceedings: Workshop On Case-based Reasoning (DARPA)*, San Mateo, California: Morgan Kaufmann, 1988.

[20] Stonier, H, and L. Marshall, "Moving To Problem-Based Learning in the NZ Engineering Workplace," *Journal of Workplace Learning*, 14, (5), 2002, pp. 190–197.

[21] Petroski, H., *Invention By Design: How Engineers Get From Thought To Thing*," Cambridge, Massachusetts: Harvard University Press, 1996.

[22] Jonassen, D.H., and P. Henning, "Mental Models: Knowledge In The Head And Knowledge In The World," *Educational Technology*, *39* (3), 1999, pp. 37–42.

[23] Hutchins, E., *Cognition In The Wild*, Cambridge, Massachusetts: MIT Press, 1995.

[24] Brown, A.L., D. Ash, M. Rutherford, K. Nakagawa, A. Gordon, and J.C. Campione, "Distributed Expertise In The Classroom," in G. Salomon, ed., *Distributed Cognitions: Psychological And Educational Considerations* (pp. 188-228), Cambridge: Cambridge University Press, 1993.

[25] Pea, R.D., "Practices Of Distributed Intelligence and Designs For Education," in G. Salomon, ed., *Distributed Cognitions: Psychological And Educational Considerations* (pp. 47-87), Cambridge: Cambridge University Press, 1993.

[26] Konradt, U., "Strategies Of Failure Diagnosis In Computer-Controlled Manufacturing Systems," *International Journal of Human Computer Studies, 43*, 1995, pp. 503–521,

[27]Bereiter, S.R., and S.M. Miller, "A Field Study of Computer-Controlled Manufacturing Systems," *IEEE transactions on Systems, Man, and Cybernetics, 19*, 1989, pp. 205–219.

[28] Jonassen, D.H., and W. Hung, "Learning To Troubleshoot: A New Theory-Based Design Architecture," *Educational Psychology Review,* (in press).

[29] Simon, D.P., "Information Processing Theory of Human Problem Solving," in D. Estes, ed., *Handbook Of Learning And Cognitive Process*, Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1978.

[30] Jonassen, D.H., "Using Cognitive Tools To Represent Problems," *Journal of Research in Technology in Education, 35* (3), 2003, pp. 362–381.

[31] Savelsbergh, de Jong, and Ferguson-Hessler, "Learning With Multiple Representations," M.W. van Someren, P. Reimann, H. Boshuizen, and T. de Jong (Eds.), Amsterdam: Pergamon Press.

[32] Zhang, J., "The Nature Of External Representation In Problem Solving," *Cognitive Science,* 21 1997, pp. 179–217.

[33] Bransford, J.D., and D. Schwartz, "Rethinking Transfer: A Simple Proposal With Multiple Implications," *Review of Research in Education, 24*, 1999, pp. 61–100.

[34] LaPlaca, M C., W.C. Newstetter, and A.P. Yoganathan, "Problem-Based Learning In Biomedical Engineering Curricula," *Proceedings - Frontiers in Education Conference*, 2, 2001, F3E/16-F3E/21 (IEEE cat n 01CH37193).

[35] Cline, Matthew, J. Powers, and J. Gary, "Problem Based Learning In A Chemical Engineering Undergraduate Laboratory," IEEE Frontiers in Education, 1997, pp. 350–354.

[36] Armarego, J., "Advanced Software Design: A Case In Problem-Based Learning," IEEE Computer Society, *Proceedings of the 15th Annual Conference on Software Engineering Education and Training*, (2002).

[37] Mitchell G.G., and J.D. Delaney, "An Assessment Strategy To Determine Learning Outcomes In A Software Engineering Problem-Based Learning Course," *International Journal of Engineering Education*, 20 (3), 2004, pp. 494–502.

[38] Van Kampen, P., C. Nanahan, M. Kelly, E. McLoughlin, and E. O'Leary, "Teaching A Single Physics Module Through Problem Based Learning In A Lecture-Based Curriculum," *American Journal of Physics*, 72 (6), 2004, pp. 829–834.

[39] Denayer, I., K. Thaels, J. Vander Sloten, and R. Gobin, "Teaching A Structured Approach To Design Process For Undergraduate Engineering Students By Problem-Based Education," *European Journal of Engineering Education*, 28 (2), 2003, pp. 203–214.

[40] Brodeur, D., P.W. Young, and K.B. Blair, "Problem-Based Learning In Aerospace Engineering Education," *Proceedings, American Society for Engineering Education Annual Conference and Exposition*, 2002.

[41] Ellis, A., L. Carswell, A. Bernat, D. Deveaux, P. Frison, V. Meiselo, J. Meyer, U. Nulden, J. Rugelj, and J. Tarhio,, "Resources, Tools, And Techniques For Problem-Based Learning In Computing," *Association for*

Computing Machinery: Report of the ITiCSE'98 Working Group on Problem Based Learning, 1998.

[42] Cirstea M, "Problem-Based Learning (PBL) In Microelectronics," *International Journal of Engineering Education*, 19 (5), 2003, pp. 738–741.

[43] McIntyre, C., "Problem-Based Learning As Applied To The Construction And Engineering Capstone Course at North Dakota State University," *Proceedings - Frontiers in Education Conference. 2*, 2003, F2D/1-F2D/6 (IEEE cat n 02CH37351).

[44] Zywno, M.S., and D.C.I. Kennedy, "Integrating the Internet, Multimedia Components, And Hands-on Experimentation Into Problem-Based Control Education," *Proceedings, Frontiers in Education Conference*, 1, 2000, IEEE, Piscataway, New Jersey, 00CB37135. T2D-5-T2D-10.

[45] Chandrasekaran, B., "Design Problem Solving: A Task Analysis," *AI Magazine*, 11 (4), 1990, pp. 59–71.

[46] Flesher, J.W., "An Exploration Of Technical Troubleshooting Expertise In Design, Manufacturing, and Repair Contexts," *Journal of Industrial Teacher Education, 31* (1), 1993, pp. 34–56.

[47] Johnson, S.D., "A Description Of Experts And Novice Performance Differences On Technical Troubleshooting Tasks," *Journal of Industrial Teacher Education, 26*, 1989, pp. 19–37.

[48] Jonassen, D.H., M.A. Schmidt, W. Miller, and G. Neumeyer, "A Problem-Based Introduction To Nuclear Sciences," *American Society of Engineering Education*, Portland, Oregon, (June, 2005).

[48] Jonassen, D.H., "Learning To Solve Problems Online," in C. Vrasidas and G. Glass, eds., *Distance Education And Distance Learning* (pp. 75–98) Greenwich, Connecticut: Information Age Publishing, 2002.

[49] Jonassen, D.H., "Designing Research-Based Instruction For Story Problems," *Educational Psychology Review, 15* (3), 2002, pp. 267–296.

[50] Jonassen, D.H., and W. Hung, "Learning To Troubleshoot: A New Theory-Based Design Architecture," *Educational Psychology Review,* (in press).

[51] Jonassen, D.H., M.A. Schmidt, W. Miller, and G. Neumeyer, "A Problem-Based Introduction To Nuclear Sciences," *American Society of Engineering Education*, Portland, Oregon, (June, 2005).

[52] Wankat, P.C., *The Effective, Efficient Professor*, Boston, Massachusetts: Allyn & Bacon, 2002.

[53] Johnson, D.W., and R.T. Johnson, "Cooperative Learning Returns To College: What Evidence Is There That It Works?," *Change, 30* (4), 1998, pp. 26–36.

AUTHORS' BIOGRAPHIES

David Jonassen is a Distinguished Professor of Education at the University of Missouri where he teaches in the areas of Educational Psychology and Learning Technologies. Since earning his doctorate in educational media and experimental educational psychology from Temple University, Dr. Jonassen has taught at the Pennsylvania State University, University of Colorado, the University of Twente in the Netherlands, the University of North Carolina at Greensboro, and Syracuse University. He has published 28 books and numerous articles, papers, and reports on task analysis, instructional design, computer-based learning, cognitive tools, and problem solving. His current research focuses on models and methods for engaging and supporting problem solving. He is Director of the Center for the Study of Collaborative Problem Solving.

Address: Department of Educational Psychology and Learning Technologies, 221C Townsend Hall, University of Missouri, Columbia, Missouri 65202; telephone: (+1) 573.882.2832; fax: (+1) 573.884.4944; e-mail: jonassen@missouri.edu.

Johannes Strobel is assistant professor in the Educational Technology programme and member of the Centre for the Study of Learning and Performance at Concordia University, Montreal, Canada. After studying Philosophy, Religious Studies, Psychology, and Information Science, he finished his Ph.D. at the School of Information Science and Learning Technologies (SISLT) at the University of Missouri-Columbia, USA. He is focusing his research and teaching on the intersection between learning and technology. He is interested in the use of computers as cognitive tools, participatory design, collaborative authoring of hypertext systems, problem solving in ill-structured and complex domains, and in social computing. He is drawing from theories of conceptual, domain-specific reasoning, and employs qualitative and design-based research methodologies.

Address: Educational Technology Programme, Concordia University, Rm LB-78-10, 1455 DeMaisonneuve West, Montreal, Canada, H3G 1M8; telephone: (+1) 514.848.2424 extension 7338; fax: (+1) 514.848.4520; e-mail: jstrobel@education.concordia.ca.

Chwee Beng Lee currently works as a lecturer at the Nanyang Technological University (Singapore), preparing pre-service teachers and conducting research at the Learning Sciences Lab. Her research interests include systemic thinking for K-12 schools and the development of computer-mediated learning environments for conceptual change and problem solving. She is a Doctoral Candidate in Learning Technologies at the University of Missouri-Columbia.

Address: 1 Nanyang Walk, Singapore 637617; telephone: (+65) 679.03285; fax: (+65) 689.68038; e-mail: cblee@nie.edu.sg.

Appendix

Interview Protocol

The purpose of this interview is to learn about the kinds of problems that engineers solve in the workplace. As an engineer, you perform a variety of jobs or tasks for the company that you work for. Most of those involve problem solving. If we can better understand how engineers work and the kinds of problems that they solve in the workplace, we can better prepare new engineers in college and university engineering programs. Think back on your experience and recall a typical problem that you have solved (or are currently solving). We would like for you to tell us about it. But first, we need some background information.

Interviewee Background Information

What engineering degrees have you earned? Where and when?

Do you still practice in that field, or have you migrated in your task orientation?

What department/unit/section are you employed in?

How long have you worked as an engineer for this company?

What is your current job title? What is your current range of responsibilities?

What other engineering positions have you held with this company or with any other companies?

Company Background Information

Business Type: What kind of company, agency, or organization do you work for? (private industry, state agency, federal agency, military)

Size: How many employees are in your department? Location?

How many other professional, technicians, or other employees are in your department, section/unit?

Project information

For the last project that you completed, what was the main objective(s) of the project?

Who identified the need for the project?

What were the deliverables of the project? Or what was the end product? (Was this a design project or did it also include implementation and further responsibilities)

What were the main tasks that were required of you in order to complete the project? (List them based on priority)

Have you ever worked on similar projects before? When? How often?

What major constraints or functional requirements were there for the completion of this task (funding, deadlines, personnel, legal, corporate policy, etc.)? How strictly defined were they? How much did those constraints affect your solution?

In hindsight, were there any aspects of the project that could have been improved? If so what impact would that have had on the project (quicker solution, less money spent, etc.)

Was the project considered a success? What criteria were used to determine the success?

Task Completion / Solution Development Questions

What was the objective of this task?

What were the deliverables of the task? Or what was the end product?

Were they specified or left open ended?

Did the task require the design of a new product, procedure, model, or system?

Did the task require trouble shooting a problem?

Did the task require the decision between two pre-defined alternatives?

Did the task require the building of a model or prototype?

What major constraints or functional requirements were there for the completion of this task (funding, deadlines, personnel, legal, corporate policy, etc.)? How strictly defined were they? How much did those constraints affect your solution?

What other employees or non-employees did you work directly with in the completion of this task? What were their roles?

How did you know what you needed to do to complete the tasks or develop the solution?

Did your boss specify it?

Previous experience from similar tasks?

Previous experience from dissimilar tasks?

Education?

Coworkers?

Other?

How did you represent the task (formulae, prototype, model, functional description)?

How did you go about determining possible solutions? What methods of analysis where used? (functional analysis, brainstorming, decision analysis)?

Were alternative solutions considered? If so what criteria were used to determine the best solution?

Was the solution widely accepted or was it controversial?

To what degree was the task completed?

How have you documented the task and your solution?