Training in Troubleshooting Problem-Solving: Preparing Undergraduate Engineering Students for Industry

Mr. Presentacion Rivera-Reyes, Utah State University - Engineering Education

Presentacion Rivera-Reyes is currently a graduate teaching assistant and a PhD student in the Engineering Education Department at Utah State University. He formerly held a position as Professor of Telecommunication Engineering at Technological University of Honduras. He received his B.S. in Electrical Engineering from the National Autonomous University of Honduras. He has experience in the telecommunication industry where he worked training engineers and technicians in high-speed transmission system for backbone networks.

Mr. Raymond Edward Boyles, Technology and Engineering Education

Raymond Edward Boyles has an associate's in specialized, avionics, May 1992, from Pittsburgh Institute of Aeronautics, and a B.S., information technology, May 2008, and M.S., technology education, Aug. 2009, from California University of PA. Most Relevant Papers: Santoso, H., Boyles, R., Lawanto, O. & Goodridge, W. (2011). A Preliminary Study of Conducting Semi-Structured Interview as Metacognitive Assessment in Engineering Design: Issues and Challenges. American Society for Engineering Education. Presentations, five most relevant: TeleRobotics: The Internet, a Physical Sensor 73rd Annual Conference Program, ITEEA 2011. Dr. Kallis, J., Boyles, R. Implementation of Microgravity Experiments in the Classroom 73rd Annual Conference Program, ITEEA 2011 Teaching STEM Concepts with Agile Robotics. Dr. Kallis, J., Boyles, R. 73rd Annual Conference Program, ITEEA 2009. Tremaux's Algorithm with Recursion 73rd Annual Conference Program, ITEEA 2009. "Normalcy" 2008 Intercollegiate Art Show, Best of Show, California University of PA, April 3rd, 2008. "Die Verfeinerung von Sustainability" Create the Future Design Contest, NASA Tech Briefs and SolidWorks, Oct. 14, 2008. "Teaching by strong interest through analogous Java Software," 56th Annual TEAP Conference. Harrisburg, Penn., Nov. 6-7, 2008. "G.S.Ohm: Electrical Innovation in Industry." Poster presentation at National Collegiate Honors Council Conference, Philadelphia, Penn., Nov. 17, 2006. "Societas crescit: Invention and Innovation through Team Teaching." Panel presentation at National Collegiate Honors Council Conference, Philadelphia, Penn., Nov. 16, 2006. Major Collaborators: Prof. John R. Kallis (California University of Pa.) Graduate Advisor: Prof. Gary Stewardson (Utah State University). Military: U.S. Army, Fort Carson, Colo. Unit Administrative Specialist (Dec. 1998-April 1999), duties: Created and maintained databases for 231 soldiers; responsible for processing all information on incoming and outgoing personnel databases implementation saved 18% manpower. Nuclear Chemical Biological Specialist (Jan. 1998-Dec. 1998), duties: Maintained, overhauled and calibrated all chemical equipment; trained soldiers in how to React in certain NBC situations. Fire Control Repairer, duties: Served as a shop foreman; supervised eight personnel which were responsible for repairing electronic fire control equipment; repaired all electronic equipment on the M!A! and Bradley tanks; and maintained records of cost and requisitioned funding from battalion. Saved 65% in cost by performing office computer repairs. Military Achievements: Two Army Achievement medals, Army Good Conduct Medal, National Defense Service Medal, Army Lapel Button, Army Service Ribbon, Expert Marksmanship Qualification Badge with Grenade Bar, and Marksman Marksmanship Qualification Badge with Rifle Bar. Work Experience: Utah State University, Jan. 2010 to present, instructor for ETE 1020 energy, power, transportation systems control technology exploration of the concepts and processes relating to the control and automation (both hard and programmable) of technical systems in the areas of energy and power, transportation, and agricultural and related biotechnologies. California University of PA, Jan. 2008 to May 2009, Teaching Assistant. Assisted the professor in class preparation, lesson plans, and distribution of materials Also gain teaching experience by lecturing the class section which deals with programming robots. Managed a laboratory, which allowed students to complete experiments. AT&T Broadband, Pittsburgh Penn., May 2000 to Dec. 2002, Head end Technician, responsible for all aspects of high speed data, telephony and cable operations, hybrid fiber to coax transmissions, programming in Visual Basic, C++, Java scripting, and M.S. Office. Experience with systems such as Cheetah, Path Tracks, and Cornerstone. Access Bandwidth Technician, responsible for

©American Society for Engineering Education, 2013

implementing and maintaining telephony network systems for the Pittsburgh and surrounding areas. Data basing and systems programmer for repairs on nodal analysis and problem solving, Heilig-Meyers Furniture Company, Richmond, Va., Aug.1994-Nov. 1994, Network/Telephone/Alarm System Installer, traveled throughout the United States to install computer networks, telephone and alarm systems, which also included maintenance and repair. Guardian Glass Corporation, Floreffe, Penn., Electrical Maintenance Technician, Nov. 1993-June 1994, installed PLC systems (Allen-Bradley); repaired and maintained laser systems, photo helix, pyrometers, and integrated control systems for network operations. WLTJ-WRRK radio station, installed and calibrated all transmission equipment for this radio station.

©American Society for Engineering Education, 2013

Training in Troubleshooting Problem-Solving: Preparing Undergraduate Engineering Students for Industry

Abstract

Training in troubleshooting in industry tends to take place from the school of hard knocks, by trial and error, on-the-job training, and gradually from the experience of solving problems as they occur, with no well-design program of instruction. This is relatively ineffective and it does little to develop self-confidence. Troubleshooting problem-solving is a higher-level cognitive process that could range from the identification, symptoms to determine, and action required to fix a problem. The knowledge and cognitive process skills needed for troubleshooting are becoming increasingly valuable. By developing problem-solving skills, engineers will become more adept at troubleshooting problems. Researchers have done studies on how to improve the troubleshooting performance of technicians in strategizing to solve a problem.

Unfortunately, many engineers who decided to go to industry do not have developed troubleshooting skills. One may theorize that a lack of troubleshooting skill is a result of the lack of practical experience and understanding of equipment in an engineering students' educational preparation. There may also be a lack of the faculty's confidence in instructing students in using such open-ended experiences. To date, much of the research has not been implemented as a part of the curriculum of technical engineering careers.

This article reviews and synthesizes more than 30 studies from 20 years (1987-2007) of research in troubleshooting problem-solving. The goals of the article are fourfold: First, to introduce the concept of troubleshooting problem-solving. Second, to present a description of problem-solving skills needed to succeed in troubleshooting. Third, to describe strategies for instruction of engineers and technicians. We conclude that troubleshooting problem-solving should be implemented as part of engineering curricula to build on students strengths to enhance their skills to succeed in the performance of troubleshooting process in industry.

1. Definition of troubleshooting problem-solving

In 1991, Perez [23] described troubleshooting as a task that deals with problem-solving skills that are specific to a domain such as computer programming, engineering, biology, medicine, or psychology. Further, he described the task of troubleshooting is to locate the problem or malfunction in a system that is not working properly and then to repair or replace the faulty part or component. The level of details at which the troubleshooter must identify the source of the malfunctions depends on her or his role and the characteristics of the troubleshooting situation, e.g., the complexity of the system, the time available for diagnostic, and the nature of the malfunction. Successful completion of the task will depend on the actions required and on various skills [23].

In 1998, Randall MacPherson [5] defined troubleshooting as a subset of technological problem-solving, both of which share common and unique characteristics. All problems contain an initial or "what is" state, a solution path, and a goal or "what is desired" state. He indicated that "…the question of how to progress from a given starting situation to a desired end situation is usually the essence of each technical problem." While the study was specifically designed to address constructs of technologically based problems, MacPherson's statement could also easily apply to all manner of problem-solving (i.e., technological, social/personal, or natural/ecological). MacPherson and Jonassen [16] agreed that not all problems have a technological component. For example, a drug or spouse abuser may be faced with a multitude of personal problems, perhaps partially attributable to a poor selection of resolution options. A company owner may be faced with problems of meeting deadlines or payroll perhaps as a result of economic factors. In each of these instances, while a technological component could be present, the problems are primarily non-technological in nature.

Schaafstal, Schraagen, and Van Berlo [27] explained that troubleshooting is predominately a cognitive task that includes the search for likely causes of faults through a potentially enormous problem space of possible causes. In addition to fault detection or fault diagnosis, troubleshooting usually involves the repair or replacement of the faulty device. The emphasis in troubleshooting problem is on fault diagnosis, which involves a search of the components of the system that are producing substandard outputs. Troubleshooters then search for actions that will efficiently eliminate the discrepancy [1]. Medical diagnosis and treatment require more than troubleshooting. Although the diagnosis process is primarily troubleshooting, medical problem solving exemplifies diagnosis-solution problems, because selecting the best solution is more complex and ill-structured.

Ross [25] developed a standardized methodology to determine the effectiveness of troubleshooting problem-solving in college-students in the Information Technology (IT) field. He stated that problems can be as simple as a toolbar disappearing in a word processing program, or as complicated as being unable to access certain nodes on a network. In his research, he mentioned that the process of fixing problems is often referred to as troubleshooting: a process which ranges from identification of a problem, uncovering the symptoms, and implementing the action required to fix the problem [27]. Because troubleshooting in general is as much about making appropriate observations and testing as it is about repairing devices, IT troubleshooters must have multiple kinds of knowledge about the IT domain [25]. This is true for every troubleshooter in a specific domain, i.e., engineering troubleshooters must have multiple kinds of knowledge about engineering domain fields [28]. A troubleshooter develops a systematic approach by representing the problem at multiple level of abstraction. When considered in this manner, it is clear that troubleshooting is more than a merely a series of decisions: it is a process requiring system knowledge, conceptual knowledge of how a system is supposed to function, an understanding of effective problem solving procedures, and an ability to manipulate effectively the problem space to minimize extraneous information [15].

The term *troubleshooting problem-solving* is typically associate it with repair, replace, and diagnosis. Often, troubleshooting is associated with the repair of a mechanical or electronic device. Whether troubleshooting a faulty device, a faulty refrigeration system in a supermarket, a car that will not start, a patient with an unknown disease, or communications problems in a meeting, troubleshooting attempts to isolate fault states in a system and repair or replace the components to reinstate the system to normal functioning [16, 17]. However, organizational staff personnel such as employee-relations managers, customer-relations specialists, consumer advocates, public-relations specialists, and human-resource directors are also troubleshooters [16]. A medical doctor is an example of troubleshooter who diagnoses an illness based on a patient's symptoms; psychotherapists are also troubleshooters, attempting to isolate the cause of mental problems.

Jonassen [16] mentioned that troubleshooting problems are moderately ill-structured. There is usually a finite problem or set of problems that are causing difficulties. He explains that troubleshooting problems:

1. Appears ill-defined because the troubleshooter must determine what information is needed for problem diagnosis (e.g., which data about the electrical and fuel system are needed in troubleshooting a car that will not start)
2. Require deep-level understanding of the system (e.g., how do electrical, fuel, and mechanical systems interact)
3. Usually possess a single fault state, although multiple faults may occur simultaneously (e.g., faulty battery, clogged injector)
4. Have known solutions with easily interpreted success criteria (e.g., part replacement leads to system restart)
5. Require learners to make judgments about the nature of the problems
6. Vary significantly in terms of system complexity (age, manufacturer, engine size, reliance on computer controls in the automobile)

Jonassen stated that a major criterion of troubleshooting problems is fault identification and efficiency of fault isolation [16]. He separated this from diagnosis-solution problems, indicating that a criterion is different in diagnosis, based on the strategies used and the effectiveness of the treatment. However, most of the other researchers in troubleshooting problem-solving have agreed that the diagnosis is also a part of troubleshooting problems.

2. Engineers as a troubleshooter problem-solvers

In 2008, engineers held about 1.6 million jobs in the U.S., 36 percent of which were in manufacturing, 30 percent were in the professional, scientific, and technical services industries, primarily architectural, engineering, and related services [8]. Many engineers also work in the construction, telecommunications, and wholesale trade industries. They need to have problem-solving skills because engineers are required to troubleshoot problems. As a consequence, product engineers in the 21st century must adopt problem-solving and independent critical-thinking skills to their work. Instead of focusing on just the mechanical aspects, design solutions must be sustainable, user-centered, responsible, and appropriate [31].

Diagnosing the reasons for malfunctioning equipment and machinery is an important facet in our industrial economy. This nation's quality of life is dependent upon the ability of our workforce to identify and solve technical problems. The steadily growing service sector of the nation's economy is a prime example of the need for problem-solving abilities in the workforce. It is becoming increasingly difficult to maintain an adequate workforce skilled in repairing equipment and machinery. The knowledge and cognitive process skills needed for troubleshooting and repair are becoming increasingly valuable. The problem, however, lies in our lack of understanding of the knowledge and skills that are required to perform the complex task of troubleshooting faulty equipment [10].

Unfortunately, many engineers who decided to go to industry do not have developed troubleshooting skills. One may theorize that a lack of troubleshooting skill is a result of the lack of practical experience and understanding of equipment in an engineering students' educational preparation. There may also be a lack of the faculty's confidence in instruction students in using such open-ended experiences [33]. By developing problem-solving skills, it is logical that engineers will become more adept at troubleshooting problems. Research on troubleshooting problem-solving has shown that experienced engineers differ from poor engineers' primarily in their strategizing to solve troubleshooting problems. Also, researchers have done studies on how to improve the troubleshooting performance of technicians in strategizing to solve a problem [12, 13]. However, to date, much of the research has not been implemented as a part of the curriculum of technical engineering careers. Therefore, it is necessary to focus on how to implement the instruction of troubleshooting problem-solving as an essential element in engineering curricula.

3. Troubleshooting problem-solving skills.

According to the Career Guidance and Students Welfare Department of Bharathiar University [3], *skill* or *ability* is defined as the learned capacity to carry out pre-determined results often with a minimum outlay of time, energy, or both. Skills are often divided into domain-general and domain-specific skills. For example, in the domain of work, examples of general skills would include time management, teamwork and leadership, self motivation and others, whereas domain-specific skills would be useful only for a certain job. Skill usually requires certain environmental stimulus and situations to assess the level of skill. A problem-solving skill includes the ability to recognize and define problems, invent and implement solutions, and track and evaluate results.

Perez [23] concluded that are three important general abilities to troubleshoot: repair and replacement of components, performance of diagnostic tests, and use of a general strategy. Perez summarized the literature on troubleshooting as follows:

> With respect to repair and replacement of components, ineffective troubleshooters demonstrated a lack of elementary knowledge and were poor in executing and verifying the results of their work. When performing tests, poor troubleshooters made fewer tests and more useless tests and were more inconsistent in their considerations of test difficulty. The strategic behavior of poor troubleshooters was characterized by incomplete and inappropriate use of information, ineffective hypothesis generation and testing, and generally less strategic flexibility. (p. 505)

If we accept the hypothesis that the ability to troubleshoot is comprised of these three abilities, then a troubleshooting training program must provide training that would teach novices to develop skills in all three areas. The training of repair and replacement procedures and test performance appears to be rather straightforward.

Schaafstal, Schraagen, and Van Berl [27] showed, in integrated circuit and papermaking fields, that expert troubleshooters use a structured approach consisting of a number of steps in a particular order, deviating minimally from a normatively optimal top-down and breadth-first method. The function of this approach is primarily memory management. Particularly, the control is essential for preventing working memory overload, which occurs when students lose track of where they were in the problem-solving process when unexpected problems were encountered while troubleshooting. For example, in testing activities, students may encounter a problem in interpreting a signal, leaving them puzzled as to whether used measurement tool correctly or measured at the right place. This may result in taking measurements at various places or recalibrating measurement tools to insure the original measurement was correct. If the process is lengthy, it is likely that the students will have forgotten why and what they were measuring, resulting in having to repeat the troubleshooting process [27]. Novice troubleshooters, lacking a good control strategy, do not see the forest for the trees, although this may also be attributable to their lack of a functional representation of the system [26, 27].

In a study by MacPherson [20], it was concluded that problem-solvers must have a firm understating of cognitive technical knowledge. MacPherson corroborated the conclusion based on previous investigation work of Sternberg [29] and Stevenson [30]. The cognitive technical knowledge variable ranked second only to "years of experience" in correlation strength. The inference is clear. Employers must provide technicians with the opportunity to develop, enhance, and continually update cognitive content knowledge, particularly in areas where the technology changes rapidly. It is also clear that the base of cognitive knowledge can be expanded and made more practical through years of experience. Moreover, he indicates that, because problem-solving is a process that encompasses all facets of human existence, personality characteristics such as perseverance, ingenuity, self-confidence, and patience are common to both technological troubleshooting and general problem-solving. Technological problem-solving encourages creativity, ingenuity, and inventive thought processes. Inventiveness, creativity and the ability to "think on one's feet" are characteristic of engineering behavior desired by employers [20].

MacPherson [20] and Ross [25] defined troubleshooting problem in the technological field. They agreed that technological problem-solving (or troubleshooting problem-solving) is a key tenet of higher-order thinking [18, 32] and that technological problem-solving is, by definition, rooted in real-life or authentic domains [4, 5].

Jonassen [14] stated that troubleshooting skill requires: system knowledge (how the system works), procedural knowledge (how to perform problem-solving procedures and test activities), and strategic knowledge (strategies such as search-and-replace, serial elimination, and space splitting [24]). These skills are integrated and organized by the troubleshooter's experiences. The troubleshooter's mental model consists of conceptual, functional, and declarative knowledge, including knowledge of system components and interactions, flow control, fault states (fault characteristics, symptoms, contextual information, and probabilities of occurrence), and fault-testing procedures [14]. The primary differences between expert and novice troubleshooters are the amount and organization of system knowledge [11, 12]. Troubleshooting requires an integrated understanding of how the system works, which is best taught through functional flow diagrams [10].

Woods [33] mentioned there are five key skills inherent in solving troubleshooting problems: knowledge about a range of process equipment, knowledge about the properties, safety and unique characteristics of the specific process conditions where the trouble occurs, "system" thinking, and people skills. He states that for general problem-solving, one of the most critical skills is to identify which evidence is significant and how it relates to appropriate hypotheses and conclusions. Concerning the importance of knowledge about process equipment, he referred to expert and novice troubleshooters as skilled and unskilled troubleshooters, respectively. Woods [33] states that differences between skilled and unskilled troubleshooters have more to do with their repertory of experiences than differences in general problem-solving skills. It is their knowledge about process equipment, common faults, typical symptoms and their frequency that is of vital importance. But troubleshooter effectiveness depends primarily on the quality of knowledge that relates symptom to cause, and the relative frequencies of the symptoms and the likely causes. Woods also mentions that interpersonal skills needed must include good communications and listening, building and maintaining trust, and understanding how biases, prejudice, and preferences lead to interpersonal differences in style [33].

Based on the findings of Perez [23], Jonassen [15], MacPherson [20], and Woods [33]., critical-thinking skills appear to be important in the process of solving problems, where potential solutions are evaluated in an iterative cognitive procedure until a decision is made for the most useful and practical problem-solving strategy. These findings reinforce the notion that the components of critical-thinking skills identified by Glaser in 1941, i.e., recognition of problems, gathering of pertinent information, recognition of unstated assumptions and values, comprehension and use of language with accuracy, clarity and discrimination, are relevant indicators of technological problem-solving skills.

The process of troubleshooting requires an integrated ability to collect, process, and evaluate external and internal information. A correct fault solution may be obtained after numerous iterations of incorrect hypothesis evaluation, or information interpretation. Conversely, an incorrect judgment may result from a single failure to correctly interpret acquired information or evaluate and generate a hypothesis. This implies that an important element in the process is the ability to continue the cycle of processes represented in the Technical Troubleshooting Model, and to verify proposed solutions [10]. It is interesting to note that Johnson and Perez are the only researchers who use the word *hypothesis* to define troubleshooting problem-solving.

4. Strategies to solve troubleshooting problems

Perez [23] stated that strategies used in troubleshooting vary from simply starting with the components nearest to the troubleshooter and tracing back to the source of the problem. The generation of a hypothesis is based on the identification of the symptoms of malfunction and knowledge of how the system functions normally, followed by tests to confirm, reject, and/or revise the hypothesis. Perez mentions that expert troubleshooters often develop and employ heuristics (i.e., rules of thumb that are useful for solving problems) such as adopting a split-half or bracketing approach or following a predetermined set of procedures. It is these troubleshooting strategies that intuitively appear to have the most promise as generic problem-solving skills.

Flesher [9] developed a study to explore electronic troubleshooting in different contexts of design, production, and repair. He made reference to the model explained by Johnson [11], the Technical Troubleshooting Model, that reflected the cognitive process flow of an engineer engaged in troubleshooting technical problem. The model is divides into two main phases (a) hypothesis generation and (b) hypothesis evaluation. In phase one the problem-solver acquires information from internal or external sources that can be used to support a representation of the problem. Following this representation, one or more hypothesis are developed that may account for the fault. In phase two, the problem solver evaluates a hypothesis generated in phase one and attempts to confirm or disconfirm the potential cause of failure [11]. During this phase, the troubleshooter determines if the potential fault is true, and if so, makes a determination and repairs the problem. Troubleshooting is often a cyclic process, with each level representing a closer approximation of the actual fault. If the fault is not identified, the troubleshooter repeats the process of hypothesis generation [10, 11].

Moreover, in the paper "Understanding Troubleshooting Styles to Improve Training Methods" [12] based on verbal protocols, Johnson stated that there are three phases that constitutes effective troubleshooting: problem representation, fault isolation, and solution verification. Based on this assumption, most troubleshooters have little problem verifying that a fault does exist in a system. The problem representation phase includes the initial evaluation of a system and any assumptions brought to the problem-solving situation. These early efforts established a broad definition of the task environment [12].

The fault isolation phase is accomplished within the framework of the hypothesis generation and testing cycle. Troubleshooters use various methods to isolate faults depending upon factors such as troubleshooter level of expertise, the type of system, and the difficulty of the problem. Johnson [11, 12, 13] and Brown et al. [2] identified five common methods:

1. Trial and error: Randomly attack any section of the system where the possible fault might have occurred (common in the performance of novice troubleshooters).
2. Exhaustive: List all the possible faults and test them one by one until the actual fault is identified.
3. Topographic: Isolate the fault through identifying a series of functioning and malfunctioning checks following the traces through the system. The topographic strategy is usually implemented in two ways, forward or backward. The forward topographic strategy starts the troubleshooting procedure at a point where the device is known to be functioning normally and then works toward the fault by following the system. The backward topographic strategy follows the same procedure but starts at the point of malfunction and then works backward to the input point [14, 21].
4. Split-half: Split the problem space in half and check the functioning condition to determine in which half the fault is located. The procedure is repeated until the potential faulty area is reduced to a single component. This strategy is efficient when the faulty system is complex and the initial problem space appears to contain several potential faults with no strong indication of where the actual fault lies [7].
5. Functional/discrepancy detection: Isolate the fault by looking for the mismatches between what is expected in a normal system operation and the actual behaviors exhibited [2]. By detecting the mismatches, the troubleshooter can identify the components where the difference is located and, in turn, isolate the actual fault.

Moreover, Johnson[13] mentioned that trial and error is a random search rarely used by experts. Exhaustive searches require little expertise, but are efficient only in situations where the component set is small. Topographic searches are directed by tracing a schematic through the system identifying a series of good/bad checks. The half/split or binary search method is an efficient means of reducing a problem space by checking for a proper condition at the middle of the remaining problem space. The process is repeated until single component remains [12, 13]. Johnson mentions that each strategy may be useful under certain circumstances. Topographic, exhaustive, and trial and error searches are selected because cognitive effort is needed. Some methods, such as half/split, are selected because they are efficient at eliminating a large number of possibilities.

The solution verification phase occurs when the troubleshooter confirms the validity of the final decision. Although it is important, not every troubleshooter performs this phase. This is based on the collected verbal protocol by Johnson, in which most troubleshooters stop troubleshooting after they think they found the fault. In addition, he found that subjects who verified their previous decision were more likely to troubleshoot successfully [13].

A study by Schaafstal et al. [27] concluded that, in general, troubleshooting problem-solving can be subdivided into four sub-tasks: formulate problem description, generate causes, test, repair, and evaluate. According to Schaafstal, the most important considerations of each subtask are to:

1. Formulate problem description: In the early stage of troubleshooting, the troubleshooter must identify not only what the system is doing wrong but also what the system is doing right. This is an important task because by determining what the system is doing right, the troubleshooter discriminates certain parts that need no further investigation.
2. Generates causes: Schaafstal mentions that a distinction can be made between familiar and unfamiliar problems [27]. In the case of familiar problems, causal hypotheses are generated mainly by a process of recognition. The more experienced the troubleshooter, the more accurate the hypothesis that might be generated based on her or his experience.
3. Test: When possible causes have been generated, they must be tested. The troubleshooter must choose the right testing method and means. Schaafstal maintained that by *testing means,* we mean all tools and equipment that can provide clarity to a technician concerning the status of the system.
4. Repair and evaluate: Once a troubleshooter has identified the correct cause of the problem, the cause must be eliminated.

Schaafstal stated that the goal of troubleshooting is to return the system to its normal working state. In doing so, the cause must be returned to the lowest replaceable unit level, and the remaining causes must be evaluated [27].

Steve Litt [19], a computer programmer, technical writer, book author, and troubleshooting course instructor, developed the Universal Troubleshooting Process (UTP), a nine-step process that consists of the following steps:

1. Prepare.
2. Make a damage control plan.
3. Get a complete and accurate symptom description.
4. Do an appropriate general maintenance.
5. Narrow it down to the root cause.
6. Repair or replace the defective component.
7. Test.
8. Take pride in your solution.
9. Prevent future occurrences of the problem.

Litt [19] mentioned that the UTP process is not right for every situation. Depending upon the circumstances, some of the steps can be combined. Steps 3, 4, 5, and 6 can loop, and it is possible to drill down from system to subsystem to subsystem, performing a complete UTP in at each step. Step 2, *Make a damage control plan*, can be steps 2, 3 or 4 depending on the circumstances. Moreover, he stated that the UTP model is highly optimized for typical technological troubleshooting because it is fast, easy to follow, adaptable, and standardized common usage. Finally, Litt declared that by using the UTP instead of complex analysis, the troubleshooter's mental workload is lessened, facilitating a speedy solution to the problem.

In his book, *Successful Trouble shooting for Process Engineers*, Woods [33] described a strategy for solving problems in six stages:

1. Engage with the problem or dilemma, listen, read carefully, and manage your stress well.
2. Analyze the data available and classify it: the goals, the givens, the system, the constraints, and the criteria.
3. Explore: build up a rich visual/mental picture of the problem and its environment
4. Plan your approach to solving the problem: analyze, manage resources, make decisions, and apply heuristics.
5. Carry out the plan: carefulness, systematic attention to detail, monitoring.
6. Check the accuracy and pertinence of your answer: analysis, communication, creativity, and evaluation.

Woods [33] called the stages a "Problem-solving process, how", and developed the strategy in solving case studies in the field of chemical engineering [6]. He used the strategy to identify the changes that occurred that resulted in malfunctioning in system. Woods explained that the hypothesis is that the symptoms arise because of some change made to the system. Therefore, the plan is identify the change, in which the basic approach is to learn to answer the right question. This approach is usually most helpful for problems that occur just after maintenance and for processes that worked well in the past, but are now malfunctioning after an alteration was made, i.e., in raw material, operating procedure, operators, weather, or in a season.

Ross and Orr [25] developed a troubleshooting methodology composed of six steps. It will be referred to herein as the DECSAR Method (Figure 1). The six steps used in the method are: define the problem, examine the environment, consider the causes, consider the solutions, act and test, and review the troubleshooting:

1. In the first step, defining the problem, troubleshooters are instructed to carefully consider what is causing a system to malfunction in order to maximize the chances of finding an effective solution.
2. The second step, examine the environment, encourages troubleshooters to make systematic observations about the system, noting both functioning and malfunctioning components and maintaining a written record of observations for future reference.
3. The third step, consider the causes, requires troubleshooters to propose a number of possible reasons for the malfunctioning system and then rank order them according to the likelihood that the problem is responsible for the malfunction.
4. The fourth step, consider the solutions, follows the same process as that for consider the causes, and ensures that a troubleshooter has different courses of actions in case the first solution is ineffective.
5. The highest-ranked solution is then used for the next step, act and test, in which the solution is implemented and verified for efficacy. During this step, troubleshooters are encouraged to compare the current functioning of the system with the notes made during the second step.
6. Finally, once the system is repaired, troubleshooters are encouraged to review troubleshooting in order to develop a more complete understanding of the system and to look for errors/shortcuts in the troubleshooting process.

Troubleshooters apply a strategy following a specific sequence of events to achieve a solution [25]. Thus, instead of learning the techniques for effectively solving problems, troubleshooters may learn to follow pre-determined steps, rather than randomly trying to solve the problem. Ross and Orr's assumption is that, regardless of the specifics of the method being used, one of the main goals of a troubleshooting methodology is to reduce the load on working memory by making the troubleshooting process more automatically [25].

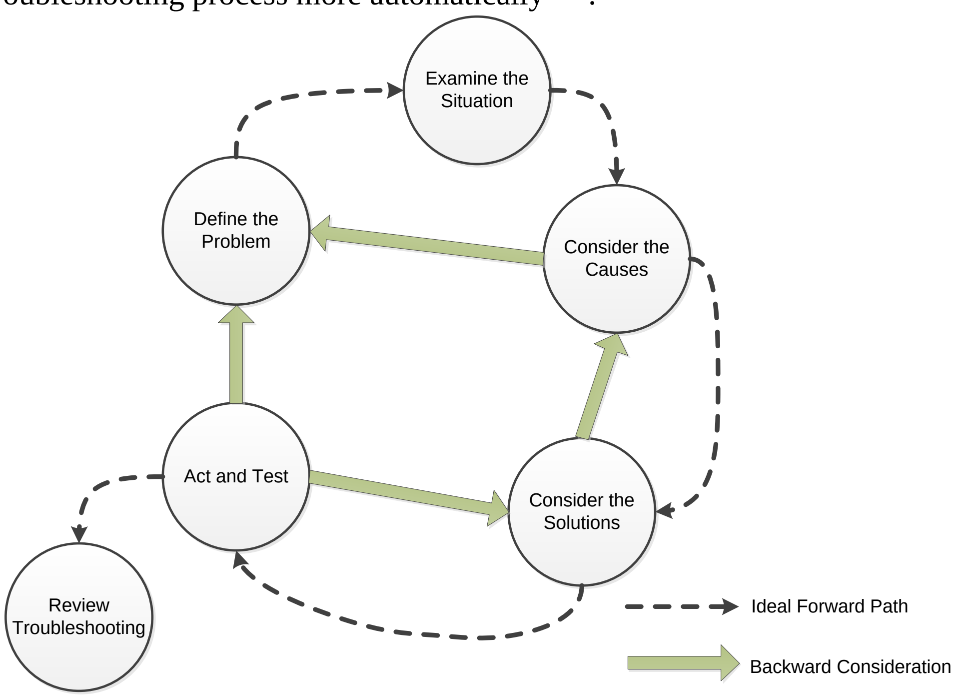

Figure 1 - DECSAR Method by Ross and Orr [25]

The different stages of the troubleshooting process are similar to those found in other methods. The specific contribution of the DECSAR Method is inclusion of a more cyclical process by means of backward steps and the explicit recommendation to use the method in an interpersonal context. The results of this study suggest that one of the major strengths of the DECSAR Method may be in its generalizability of use, which may result from reinforcement of the idea that recursive thinking (i.e., moving backwards through the process) is an effective component of troubleshooting.

5. Implications

Perez [23] posited that troubleshooting is a task which deals with problem-solving. Schaafstal [27] also defined a troubleshooting problem as a task, adding the term "cognitive." However, troubleshooting problem-solving is more that a task; it is a higher-level cognitive process that ranges from the identification of a problem, the symptoms to determine and the action required to fix a problem [25]. It is a higher-level process because troubleshooting problems are moderately ill-structured [13], with complex system including faults and numerous optional solutions in a real-world context [12].

Most researchers agree that troubleshooting problems can occur in every field, including technology, medicine, and psychology. In the case of technology, the problem is typically related to repair and replacement of faulty components of a system, and requires creativity, ingenuity, and inventive thought process. It involves activities including the collection, processing, and evaluation of external and internal information [9].

Johnson differentiates between troubleshooting and diagnosis problem-solving [12]. These terms are also used differently in literature. Diagnosis often refers to the identification of the symptoms to determinate a faulty component. Because troubleshooting problem-solving is defined as an entire process involving identification of symptoms, fault determination, and compensatory actions [27], diagnosis is part of this process and, as a consequence, a stage or step in the entire process of troubleshooting problem-solving.

Skill or ability is defined as a learned capacity to carry results in a specific domain. As a result, novice troubleshooters may learn skills to develop their expertise. Researchers have agreed that knowledge of content is fundamental in problem-solving. A troubleshooter must have the ability of *critical thinking*: recognition of the problem, gathering of pertinent information, recognition of unstated assumptions and values, comprehension and use of language, clarity, and discrimination [20]. In the case of troubleshooters in technology, computer, or engineering fields, they must continually update their cognitive content knowledge to remain current in rapidly changing.

A troubleshooter must have the skill to *follow a structured approach* to troubleshoot based on steps, in a particular order. The importance of this skill is to prevent overload working memory management [27]. Working memory overload occurs when troubleshooters have lost where they were in the problem-solving because they encountered unforeseen problems while troubleshooting. If the process of troubleshooting takes some time, there is a fair chance that troubleshooter will have forgotten why and what they were doing at all, resulting in having to back up in the troubleshooting process and consequently, overloading the working memory [27].

Woods [33] made reference to *interpersonal skills* (communication and listening skills) needed between troubleshooter and the people with whom she/he interacts in solving a problem. These include skills in communication, listening, building and maintaining trust, and building on personal uniqueness. It is important to consider interpersonal skills and factors such as willingness risk admitting mistakes, stress, distress, and environment that could affect the performance of the troubleshooters (2006). Furthermore, in engineering fields, professionals must develop skills working with teams as a collaborator or manager.

Strategies used in troubleshooting may vary depending on the components and system. The DECSAR Method developed by Ross and Orr [25] applies a specific sequence of events to achieve a solution. The method was based on the four step of the strategy outlined by Schaafstal, and the assumptions of Johnson (1995), indicating that troubleshooting process is cyclical (forward and backward), an effective component of troubleshooting. The DECSAR Method was written to be consistent with methods used in IT. However, to address concerns that troubleshooters were not properly applying their knowledge, DECSAR also included an interpersonal component to encourage troubleshooters to generalize their abilities by applying DECSAR in a different context [25].

In general, the first stages of strategy to solve troubleshooting problems stated by Woods and Litt are similar to the DECSAR Method. Because Woods's strategy was designed for engineers, he added the components of communication, creativity, and evaluation, useful tools in the field of engineering management. Similarly, the UTP method by Litt added affective-domain tasks in the process of solving-problem, indicating in step 8 "Take pride in your solution," and step 9 "Prevent future occurrences of this problem."

References

1. Axton, T. R., Doverspike, D., Park, S. R., & Barrett, G. V. (1997). A model of the information-processing and cognitive ability requirements for mechanical troubleshooting. *Int. J. Cogn. Ergon. 1(3): 245–266.*
2. Brown, J. S., Burton, R. R., Bell, A. G. (1975). SOPHIE: A step toward creating a reactive learning environment. *International Journal of Man-Machine Studies 7(5): 675–696.*
3. Career Guidance and Students Welfare. Bharathiar University, Coimbatore, India. Retrieved from www.careervarsity.com/download/Skill.pptx%2019.9.2011.ppt
4. Custer, R. L. (1995). Examining the dimensions of technology. *International Journal of Technology and Design Education, 5, 219-244.*
5. Custer, R. L., Valesey, B. G. & Burke, B. N. (2001). An assessment model for a design approach to technological problem solving. *Journal of Technology Education. Virginia. Vol. 12,* No.2, Spring 2001.
6. Darabi, A., Nelson, D., & Palanki, S. (2005). Acquisition of troubleshooting skills in a computer simulation: Worked example vs. conventional problem solving instructional strategies. *Computers in Human Behavior 23 (2007) 1809–1819.*
7. David, R. (1983). Reasoning from first principles in electronic troubleshooting. *Int. J. Man-Mach. Stud. 19: 403–423.*
8. Engineers, What They Do. (2008). *Student Scholarship Newsletter*. Retrieved from http://www.studentscholarships.org/salary/679/engineers.php?
9. Flesher, J. W., (1993). Context and Expertise: The Case of Electronic Troubleshooting. American Educational Research Association. Apr 93.
10. Johnson, S. (1987). Knowledge and skill differences between expert and novice service technicians on technical troubleshooting tasks. American Vocational Annual Convention, Las Vegas, Nevada, December 1987. Retrieved from http://eric.ed.gov/PDFS/ED290043.pdf
11. Johnson, S. (1989). A description of expert and novice performance differences on technical troubleshooting tasks. *Journal of Industrial Teacher Education, 26, 19-37.*
12. Johnson, S. (1995). Understanding troubleshooting styles to improve training methods. American Vocational Annual Convention, Denver, Colorado, December 1985. Retrieved from http://eric.ed.gov/PDFS/ED389948.pdf
13. Johnson, S. D., & Satchwell, S. E. (1993). The effect of functional flow diagrams on apprentice aircraft mechanics' technical system understanding. *Performance Improvement Quarterly, 6(4), 73-91.*
14. Jonassen, D. H. (2000). Toward a design theory of problem solving. *Educational Technology: Research & Development, 48(4), 63-85.*
15. Jonassen, D. H. (2003). Instructional design for learning to troubleshoot. *Performance Improvement, 42(4), 34–38.*
16. Jonassen, D. H. (2011). Learning to solve problems. A handbook for designing problem-solving learning environment. New York: Routledge.
17. Jonassen, D. H., & Hung, W. (2006). Learning to troubleshoot: A new theory-based design architecture. *Educational Psychology Review, Vol. 18, No.1, March 2006.*
18. Lavoie, D. R. (1991). The construction and application of a cognitive-network model of prediction problem solving. Paper presented at the Annual Meeting of the National Association for Research in Science Teaching. Lake Geneva, WI.
19. Litt, S. (2001). Troubleshooting techniques of the successful technologist. A Manual of instruction in troubleshooting. Slitt@troubleshooters.com, first edition, first printing 2001. Ordered at www.troubleshoooters.com/bookstore
20. MacPherson, R. T. (1998). Factors affecting technological trouble shooting skills. *Journal of Industrial Teacher Education. Volume 35, Number 4, 5-28.*
21. Newell, A., & Simon, H. A. (1972). Human problem solving. Englewood Cliffs, NJ: Prentice-Hall.
22. Pascarella, E. & Terenzini, P. (2005). How college affects students (Vol. 2): A third decade of research. San Francisco: Jossey-Bass.
23. Perez, R. (1991). A view from trouble-shooting. Toward a unified theory of problem solving, Chap. 7, pp. 115-153, Mike U. Smith (Ed.), Hillsdale, NY: Lawrence Erlbaum Associates.
24. Pokorny, R. A., Hall, E. P., Gallaway, M. A., & Dibble, E. (1996). Analyzing components of work samples to evaluate performance. *Military Psychology, 8(3),161-177.*
25. Ross, C., & Orr, R. (2007). Teaching structured troubleshooting: integrating a standard methodology into an information technology program. *Education Tech Research Dev, 57: 251-265.*

26. Schaafstal, A. M., & Schraagen, J. M. C. (1993). Training of systematic diagnosis: A case study in electronics troubleshooting. In *Proceedings of AI-ED '93: World conference on artificial intelligence in education*. Amsterdam: IOS Press.
27. Schaafstal, A., Schraagen, J.M., & van Berlo, M. (2000). Cognitive task analysis and innovation of training. The case of structured troubleshooting. *Human Factors, 42(1), 75-86.*
28. Shin, N., Jonassen, D. H., & McGee, S. (2003). Predictors of well-structured and ill-structured problem solving in an astronomy simulation. *Journal of Research in Science Teaching, 40(1), 6–33.*
29. Sternberg, R. J. (1986). *Critical thinking: Its nature, measurement, and improvement. Viewpoints, Information Analysis*. (ERIC Document Reproduction Service No. ED 272 882)
30. Stevenson, J. (1994). Vocational expertise. In J. Stevenson, S. Billet, G. Evans, C. McKavanagh, H. Middleton, & F. Beven (Eds.). *Cognition at work: The development of vocational expertise, National Centre for Vocational Education* (pp. 7-30). Research Ltd., South Australia.
31. Thoo, M. (2011). New engineers solve 21st century problems. *Swinburne University of Technology Magazine.* Melbourne, Australia. December 2011, Issue #14. Retrieved from http://www.swinburne.edu.au/magazine/14/252/new-engineers-solve-21st-century-problems/
32. Waetjen, W. B. (1989). Technological problem-solving: A proposal. International Technology Education Association. Reston, VA.
33. Woods, D. R. (2006). Successful trouble shooting for process engineers: a complete course in case studies. Wiley-VCH Verlag GmbH & KGaA. Weinheim, Germany.