A Formative Evaluation Research Study to Guide the Design of the Categorization Step Practice Utility (MS-CPU) as an Integral Part of Preparation for the GED Mathematics Test Using the Ms. Stephens Algebra Story Problem-solving Tutor (MSASPT)

by

ChristiAnne Ritchey

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved May 2018 by the
Graduate Supervisory Committee:

Kurt VanLehn, Chair
Wilhelmina Savenye
Yi-Chun Hong

ARIZONA STATE UNIVERSITY

August 2018

ABSTRACT

The mathematics test is the most difficult test in the GED (General Education Development) Test battery, largely due to the presence of story problems. Raising performance levels of story problem-solving would have a significant effect on GED Test passage rates. The subject of this formative research study is Ms. Stephens' Categorization Practice Utility (MS-CPU), an example-tracing intelligent tutoring system that serves as practice for the first step (problem categorization) in a larger comprehensive story problem-solving pedagogy that purports to raise the level of story problem-solving performance. During the analysis phase of this project, knowledge components and particular competencies that enable learning (schema building) were identified. During the development phase, a tutoring system was designed and implemented that algorithmically teaches these competencies to the student with graphical, interactive, and animated utilities. Because the tutoring system provides a much more concrete rather than conceptual, learning environment, it should foster a much greater apprehension of a story problem-solving process. With this experience, the student should begin to recognize the generalizability of concrete operations that accomplish particular story problem-solving goals and begin to build conceptual knowledge and a more conceptual approach to the task. During the formative evaluation phase, qualitative methods were used to identify obstacles in the MS-CPU user interface and disconnections in the pedagogy that impede learning story problem categorization and solution preparation. The study was conducted over two iterations where identification of obstacles and change plans (mitigations) produced a qualitative data table used to modify the first version systems (MS-CPU 1.1). Mitigation corrections produced the second version of the MS-CPU 1.2, and the next iteration of the study was conducted producing a second set of obstacle/mitigation tables. Pre-posttests were conducted in each iteration to provide corroboration for the effectiveness of the mitigations that were performed. The study resulted in the identification of a number of learning obstacles in the first version of the MS-CPU 1.1. Their mitigation produced a second version of the MS-CPU 1.2 whose identified obstacles were much less than the first version. It was determined that an additional iteration is needed before more quantitative research is conducted.

DEDICATION

This dissertation is dedicated to my mother, who gifted me early in life with an insatiable curiosity, by entertaining it with patience, answers, questions, trips to the library, travel, and a World Book encyclopedia set, surely beyond her means at the time. My daughter and fellow Ph.D. student, Elyse A. Ritchey whose considerable and timely scholarly advice and writing acumen were pivotal in finishing it, and my son, Leslie R. Ritchey, whose computing and design genius helped me to develop the MS-CPU and MSASPT software.

This dissertation is also dedicated to Dr. Wilhelmina Savenye, my first advisor, and dissertation committee chair. Her selfless commitment to all of her students, whether in class or advisement, was and still is quite extraordinary. I am amazed both at the depth of her involvement and at the breadth of her interests in and service to higher education. I am so very grateful and fortunate to have her as a teacher, a mentor, and now a friend.

ACKNOWLEDGMENTS

First and foremost I would like to acknowledge the inestimable contribution to the design of the intelligent tutoring system used in this study and the dissertation document itself of my dissertation chair, Dr. Kurt VanLehn. As one of his doctoral student advisees, I was always treated as an intellectual equal and prospered from all of my experiences at ASU as a result of the confidence that he showed in my efforts and the self-confidence that this environment created.

I would like to acknowledge, as well, the many facets of support that Dr. Savenye provided; academically, emotionally, and, at times even spiritually, although she probably wasn't aware of where in my psyche her words would end up.

No less important was the willingness of new faculty Dr. Yi-Chun Hong to join my dissertation committee and provide critical writing support and research insights; the guidance and support of my fellow spiritual traveler and tennis partner, Ann Deputy; and last but not least, the three GED Test Preparation students who inspired me to finish something that, quite possibly, could have made earning their GED certificates a much less troublesome and time-consuming affair, Erica Rohling, Elizabeth Marquez, and Jacqueline Chan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

Figure Page

PREFACE

Thinking retrospectively over the several decades that I have been teaching, I realize that I’ve always been about constructing tools to enable learning to take place via student discourse. It could be teacher to student; student to student, or computer to student. Building an intelligent tutoring system to enable GED students to solve story problems is the next iteration of this teaching paradigm and represents what I believe to be the most effective approach to GED mathematics instruction as it closely simulates the highly efficient (VanLehn, 2011a) one-to-one tutor-to-student relationship, is portable, and virtually on demand. The notion that learning takes place largely through discourse epistemologically places me as a social constructivist using Lev Vygotsky's Social Development Theory (1978) and Jerome Bruner's Theory of Cognitive Growth (1966). I expect that the reader of this document would almost certainly come to the same conclusion.

As a high school mathematics student in a college preparatory curriculum, I had little trouble with algebra, geometry, and trigonometry. Later, in college, I learned calculus well enough, without taking a calculus class, to earn a teaching minor in physics. I say this not as a boast, but as a point of reference because one aspect of mathematical study always gave me considerable trouble: solving story problems. When I began teaching mathematics to adults in early 2004, that same difficulty with story problems was evident in nearly all of my students and has shown itself to be empirically true over the last 90-100 years (Jonassen, 2000) with a large majority of students in America's junior high and high schools. Teaching mathematics for Arizona's largest provider of GED Test preparation, I knew it to be true throughout Maricopa County and by observation and communication with my peers, the state of Arizona as well.

Problem-solving research, and in particular, story problem-solving research to mitigate the associated difficulties has grown rapidly in the last 50-60 years, and there are now many theories and methods that have been developed to teach story problem-solving. However, the search to find a teachable and effective method for leading the mathematically challenged student to become successful story problem solvers has so far shown itself to be a very difficult task. This situation is evidenced by the great majority of people in the US who hate math and story problems, by poor test performance, and by a large number of people who don't graduate from high school and either can't pass the GED test or choose not to take it. Almost 20% of the eligible adult populations in many states are without a high school diploma or a GED certificate (GED Testing, 2014).

In this paper, I identify the findings of these various research efforts, and then demonstrate how this knowledge regarding story problem solving can be innovatively synthesized to create the architecture and pedagogy that draws us closer to that teachable and effective means of enabling more positive outcomes for students engaged in solving story problems. Expected contributions will be in the field of mathematics education, where a unique pedagogical perspective will be employed, and in the field of educational technology, where a traditional example tracing intelligent tutoring system architecture, named Ms. Stephens Algebra Story Problem-solving Tutor (MSASPT), employs a comprehensive set of user determinant scaffolding utilities that provide several pathways to understanding for the student.

My work on the MSASPT intelligent tutoring system began in the winter semester of 2011 at Arizona State University. It incorporates five major, research supported, pedagogical steps on its way to the numerical representation of the solution to a story problem. As a neophyte programmer in the Flash environment, I have completed the first two steps. Each of the three remaining steps is pedagogically complete and at least skeletally programmed, but still needing a considerable time investment in completing the programming. Therefore, based on the schema-theoretic notion that a student's ability to retrieve a particular schema during the problem-solving process is critically contingent on correctly categorized problems, and with the blessing of my distinguished dissertation committee, I have limited the scope of this investigation to the first step in the MSASPT problem-solving process: explicit categorization of story problems. The formative evaluation research on the categorization step is expected, again, to yield contributions in both mathematics education and intelligent tutoring systems on the same basis as the entire MSASPT as noted above.

In Section I of the Introduction, I will present several arguments for the importance of teaching GED Test Preparation students how to become effective story problem solvers. In Section II, I will relate the problems and situations that have contributed to the difficulties that many of these students have had in solving story problems then explicate how each is mitigated by the MSASPT in Section IV. In Section III, a comprehensive review, from a historical perspective, of the theories and instructional methods that have been advanced to alleviate problem-solving difficulties will be presented. Section IV, somewhat briefly, is a description of the pedagogical considerations and learning enhancement features of the MSASPT intelligent tutoring system that uniquely address the continued difficulty that story problem-solving presents to the adult GED Test takers and the general scholastic population as well. Section V will explicate the role that categorization that categorization plays in solving story problems and its importance to the overall problem-solving task. In Section VI an in-depth review of the various methods that have been used to enable students to categorize word problems, both implicitly and explicitly, will be presented. The final section (VII) will present the theoretical framework used to design the categorization step and a description of the processes in the step that were derived from the framework.

CHAPTER 1

INTRODUCTION

Section 1: The importance of the ability to solve story problems…

“Education is not only about knowledge. It is about inspiration. It is about passion. It is about the belief that what we do in life matters. It is about moral choice. It is about taking nothing for granted. It is about challenging assumptions and suppositions. It is about truth and justice. It is about learning how to think” (Hedges, 2015). This most powerful description of education, in each of its elements, speaks to the humanity of solving problems. Virtually all learning is driven by the need to solve problems or in preparation thereof. Historically, the people that have been the most effective at solving problems have also been the most successful regardless of setting (Jonassen, 2004). Solving problems is part and parcel of the study of mathematics. It is unfortunate that, to an alarming degree, mathematics has been reduced to exercises and skills so algorithmic that mathematics become largely devoid of actual problem solving, thus misrepresenting mathematics as a discipline and shortchanging the students who study it (Wilson, et al., 1993). Robert Gagne’ stated that the essence of education is to “teach people to think, to use their rational powers, to become better problem solvers" (1980, p. 85) and mathematics should contribute extensively toward that goal. However, for any number of reasons, most of which have been in place at least since education became institutionalized, applied mathematical problem solving has remained mired in futility, if taught at all, for the great majority of students.

That we, as educators, administrators, and government continue to permit this essential goal of education to be unrealized, is a gross disservice to the students that we have been charged to educate. The consequences of such negligent behavior are situated locally, nationally, and globally and is reflected in the areas of economics, politics, security, and finance at both personal and societal levels.

The business of providing solutions to the lackluster performance of America's educational systems in this regard has been constant for at least a century. The means-ends-analysis problem-solving heuristic incorporates the solution of sub-goals to solve the problem statement. The purpose of this paper is to provide a solution to the sub-goal of raising GED Mathematics Test performance for those adults without a high school diploma. It is demonstrably true, both quantitatively and qualitatively, that the most difficult test for GED test-takers is the mathematics test. Further, the most difficult problem area in the GED Mathematics test for these students has to do with solving story problems. Providing solutions to this problem will most certainly impact solutions to other sub-goals in this domain and lead to the meta-goal of successful problem-solving for all students of mathematics.

Quantitative evidence from statistics kept by the National GED Testing Service function to confirm this difficulty in mathematics test performance. For all candidates who tested in 2013, the percentages of those achieving their jurisdiction's minimum standard score in each content area (410/800 in the United States and 450/800 in Canada) were greater than 90%, except for Mathematics. Comparatively, only 80.0% of candidates scored high enough to meet the minimum standard score in Mathematics (GED Testing, 2014). Moreover, the average test score for the second most difficult test (writing) was 508/800 and for mathematics was a distant 490/800. Consider that the 2010 U.S. Census shows more than 39 million adults (18% of the population) aged 16 and older in the United States, who weren't enrolled in any educational program, were without a high school diploma or GED credential. Moreover, an astoundingly low 2.1% of this 2010 population even bothered to take the GED test (only 1.4% passed it). I would suggest that the dismal engagement with any educational advancement opportunity, to a significant degree, is due to the fear of or discomfort with mathematics and associated story problems that were internalized as these students moved through the US educational system.

Qualitative evidence of this difficulty is derived from my decade-long experience in adult education teaching mathematics. Virtually all of the students who persisted in my classes learned to solve moderately complex first degree and simple second-degree equations. Trouble for them came when they were required to use their algebraic skills, practically, to solve various kinds of story problems. As a member of the community of adult education mathematics teachers, I can accurately state that all of us experience these same problems with most of our students. Indeed, if story problem-solving were more effectively taught, there would be a much greater percentage of GED test takers that become GED graduates and almost certainly an increase in the number of people who would be willing to take the test battery with a new-found confidence in their ability to solve story problems. This would produce a more educated workforce, greater employment mobility and increase the number of post-secondary students as they transitioned to colleges and universities; inarguably, a boon to greater economic stability in our society. According to the U.S. Department of Labor, the median annual earnings in 2008 of a person with less than a high school diploma is $8,580 less per year than a high school graduate. The over 825,000 Arizonans without a high school diploma or GED represent over $7,075,000,000 in lost annual taxable income (Arizona Department of Education, 2010).

Of course, the consequence of a significant segment of the US population that functions poorly in the domain of practical math is much more than economic. There are personal consequences as well; in the movie "Stand and Deliver," the mathematics teacher, Jaime Escalante tells his students that mathematics is the great equalizer. In that classroom context, of course, he meant that in a racially discriminating society, where the students at Garfield High School were the disadvantaged, success in mathematics would go a long way to mitigating those disadvantages. However, that appraisal can certainly apply to anyone who becomes or is mathematically accomplished, although the benefits may be more for some than others.

And social: according to research sponsored by the National Institutes of Health, the beneficial effects of higher parental educational levels on their children when the child is young are not limited to academic achievement throughout the school years, but have long-term implications for positive outcomes into middle adulthood (i.e., higher educational level, more prestigious occupations) (Dubow, et al., 2009). The ripple effect of a greater number of GED graduates goes well beyond the simple passing of a test and makes the endeavor of finding a method that enhances performance in the solution of GED Test story problems, well worth the effort.

Section 2: Factors that contribute to story problem-solving difficulty

At least as far back in human history as Aristotle and his means-ends-analysis problem-solving process, teachers, philosophers, mathematicians, and scientists have been devising methods that might provide success in situations that call for solving problems. In the present day, efforts to find a story problem-solving pedagogy that is consistently effective continue as they have for the last sixty some years. Test scores, the general abhorrence of story problems as illustrated by Gary Larsen and his Hell's Library with shelves full of story problem books, and a mathematics curriculum that only dabbles in applications such as story problems, prove the lie to any sort of progress over these many years in finding such a "holy grail".

Perhaps the inability to change the less than satisfactory trajectory of story problem-solving performance lies in the misidentification of the cause or causes of such a situation. This dissertation hypothesizes that the significant difficulties that students (and most of the rest of America's population) have with story problem-solving rooted in a largely singular cause; an inability to think in the abstract (i.e., conceptually, to make generalizations and accurately apply them). While there are several dichotomous approaches to types of thinking (system 1 vs. system 2, sequential vs. holistic, divergent vs. convergent, etc.), the oppositional to abstract thinking is thinking concretely. In its applications to the domain of mathematics, this particular thinking dichotomy among students of mathematics is situated much more closely to concrete than abstract.

In my experiences as a high school mathematics teacher and as a GED Test preparation mathematics teacher, I took pride in the excellent performances of my students as they efficiently learned to solve 1st and 2nd-degree algebraic equations. However, my pride and the excellence of most of my GED student's performances, as well as their feelings of self-efficacy in mathematics, soon disappeared as we began to consider the practical uses of what they had learned, which, of course, involved solving story problems. The almost total student mystification as to how to prepare a story problem for a solution was not only baffling but consistently present, even of the students who were doing excellent work in the other four subjects of reading, writing, science, and social studies, found on the GED test.

Why is the contrast between equation solving performance and story problem-solving performance so stark? The algorithmic solution processes used by GED students in solving moderately complex algebraic equations require much lower levels of abstract thinking than that needed in solving a story problem. Abstraction in equation solution is usually found only in the incorporation of mathematical principles used to support/explain the steps in the algorithmic process, and these can be represented concretely rather easily. When solving story problems, the required level of abstractive thought is much more significant. Virtually all existent problem-solving methods involve "understanding" as not only a first step but an entirely crucial one that determines solution success or failure. However, the term "understanding" in many topics and perhaps more so in mathematics has no widely accepted definition of what it means "to understand" (Lai, 2009). The concept of "understanding" is not only an abstractive one for those looking to find a solution to a story problem and, for all but the simplest problems, an opaque one as well.

The contrast that exists between the manifest necessity of abstract thinking abilities in solving story problems with the apparent minimal requirement for solving algebraic equations, provides inductive proof that a deficit in abstraction capability in most students is part and parcel of the difficulty that they (not just GED students) experience when attempting to solve story problems. Additionally, Warren Esty and Anne Teppo (1996) write that the inability to think in the abstract prevents many students from moving from arithmetic to algebraic thinking, where a conceptual change needs to occur, and the focus of thought must shift from numbers to operations on numbers and relationships between numbers. It seems, then, that the primary cause of story problem-solving difficulty has been identified. The solution to the problem then becomes one of providing knowable (concrete) experiences capable of bootstrapping and scaffolding students so that they can accomplish the shift in focus that Esty and Teppo write about.

Is this possible? There are a number of sources for student deficits in abstract thought. If the Ms. Stephens' Algebra Story Problem-solving Tutor can redirect even a few of them, this dissertation will have had the desired goal of advancing the field of story problem-solving instruction.

- On the whole, US educational practices and curriculum have been characterized as insufficiently challenging, diffuse in content coverage, and inhibiting the development of an in-depth understanding of mathematics concepts and relations (National Education Goals Panel. 1997).
- Cross-cultural studies (e.g., Stigler, et al., 1986; Xin, 2007) indicate that, compared to some countries (e.g., Russia, China, Philippines, Finland), U.S. textbooks don't provide adequate opportunities for students to move beyond using concrete operations so that they can begin to think symbolically or algebraically.
- David Jonassen maintains that students engage in meaningful learning when they are presented with a problem to solve. Such situations can occur anywhere and at any time, regardless of physical venue, but the only legitimate goal of an educational system is preparing students for successful engagement in their problem-solving efforts. A goal that he

feels has been largely ignored, hence the difficulty most products of the American education system have with solving problems in general and story problems in particular (Jonassen, 2005, n/a).

- Insufficient teacher expertise contributes significantly to the difficulty that GED students have with solving story problems and can be attributed to the fact that teachers themselves have difficulty solving story problems. Teachers whose expertise is outside the domain of mathematics (and this is a large majority of GED teachers) and who depend on outside sources for their curriculum, instructions, and explanations, find such information to be as opaque as it was when they were students. That the information is mostly ill-conceived, incomplete and lacking in establishing, much less reaching, clear instructional goals (Jonassen, 2000) considerably exacerbates their inability to connect with learners.
- Teachers that do have expertise in the domain of mathematics fare only marginally better as students often face severe difficulties in understanding solution methods even when they have received elaborate instructional explanations of the individual solution-steps. These difficulties may result from the fact that the solution steps are often conveyed in a rather abstract way so that learners experience difficulties in understanding the connection between a step and the change(s) that occurred in the problem state as a result of the step (Scheiter, et al., 2006).
- Any solution method that calls upon the student to perform the acutely generalizable terms of understanding or comprehension, as many story problem-solving methods do (e.g., Newell & Simon, 1972; Kintsch, 1977; Nathan, 1992; Jonassen, 2000), is going to require a non-trivial level of abstraction to get to the next step in the solution process. If this requirement is put in terms of Piaget's four stages of cognitive development, the student must be competent at the fourth stage, or formal operational. However, recent research has demonstrated that not everyone reaches the stage of formal operations (the ability to think abstractly), as Piaget once believed (Keating, 1979; Cole, 1990). Other studies have been

conducted to quantify that condition, showing that less than half of adults ever reach the formal operational stage (Dasen, 1994). This shortcoming leaves many students without the tools to understand the problem-solving processes presented to them in spite of their best efforts. Moreover, reaching the formal operational stage, for those who get there, does not occur at the same moment in everyone's intellectual development. A significant portion of the half that does reach the formal operational stage may do it at a time that is too late in our industrialized, conveyor belt of an educational system, further exacerbating the juxtaposition of needing the capability of thinking in the abstract and having it.

- Frank Lester suggests that metacognitive factors such as an inability to judge the difficulty of a task, to evaluate understanding, to collect and use information towards reaching a goal or solution, and to assess problem-solving ability, contribute significantly to problem-solving difficulty. Metacognitive actions are a driving force, along with beliefs and attitudes in whether or not a student is successful in solving problems in the domain of mathematics. (Lester, 1994). In a successful instructional strategy, the development of metacognitive skills enables students to form mental representations of the problem, select an appropriate plan for solving the problem (schema), and identify and overcome obstacles to the process (Davidson & Sternberg, 1998). Orienting and self-judging are essential metacognitive skills that are positively related to problem-solving performance, and they can be learned (Masui & DeCorte, 1999), which would point to the necessity of measuring or at least encouraging these behaviors in story problem-solving.

Different cognitive styles, student epistemological stances, affective and conative domain considerations as well as the level of the problem solver's domain knowledge also figure into the calculus of abstractive capability and story problem-solving success. There is a growing body of evidence and argument suggesting that qualitative changes in reasoning take place in the early adult years and that the changes represent a reorganization of cognitive structures (Kitchener, et al., 1989). These changes may, in conjunction with MSASPT, reduce the difficulty that most adults have in thinking conceptually and lead to story problem-solving success on the GED Mathematics test. Section IV will examine whether and how these roadblock to story problem-solving success has been mitigated by the MSASPT.

Section 3: A Historical Review of the Theories and Methods Used to Mitigate Problem-solving Difficulty:

An ability to solve problems has been a necessity, depending on your stance regarding creation, for those of the human persuasion for at least many millennia. Certainly, there is boundless evidence of earliest man solving problems with tools designed, then made by hand for a particular purpose; if Adam is your wont, he was solving huge problems right from the beginning. Actual descriptions of problem-solving methods probably began to be taught in ancient Greece and the Mayan culture around 400 BC. In Greece, Aristotle was teaching his theory of means-ends-analysis around 340 BC, which can be found in the chapter, entitled "The nature of deliberation and its objects" of his book, the *Nicomachean Ethics* (Book 111. 3, 1112b).

The formal history of the various investigations into human problem-solving processes is well-documented, and any accounts before the demise of behaviorism and the institution of cognitive science will serve very little useful purpose in a paper that is concerned with story problem-solving and intelligent tutoring systems. Certainly, there were contributions made, at least indirectly, to ITS's and story problem-solving by both Behaviorism and Gestalt Theory, and where appropriate, note will be made of such contribution. (See Appendix O for a historical account of the investigations into the problem-solving process before 1950). Gestalt psychology and its use of insight in problem-solving made the deeper investigation of the cognitive process of human problem-solving virtually impossible. Behaviorism had its advantages over how human behavior was perceived previous to its inception, but as the need for deeper explanations tied to both technological advances and the need for technologically adept people increased, so had the need for a new theory of learning. In behaviorism, if learners were met with situations where previous learning had not prepared them to understand a process, then they had no background experiences to deal with that situation until they learned a “correct” response (Edgar, 2012).

Enter the third major iteration of the psychology of human learning, Cognitive Science. Cognitive science is the scientific study of how the human mind functions. It is an interdisciplinary field that combines ideas and methods from psychology, computer science, linguistics, philosophy, and neuroscience. “The broad goal of cognitive science is to characterize the nature of human knowledge – its forms and content – and how that knowledge is used, processed, and acquired.” ("Cognitive Science", n.d.). Perhaps the dominant force that shaped the field of cognitive science was the invention, and rapid technological development, of the digital computer. The computer facilitated the field of artificial intelligence and was essential for the advancement of the field of cognitive psychology as it enabled, not only the development of ideas about ways in which humans might think, but also facilitated the demonstration of the accuracy of these thinking models through computer simulation (McClelland, 2009), or computational cognitive modeling.

The almost exponential improvement in computer computational speeds facilitated increasingly powerful computational models of cognition and an ever-increasing understanding of human thought processes that William Wundt, the founder of experimental psychology, could only have dreamt. Over the first several decades of using computational cognitive modeling to study the story problem-solving process three very powerful and thorough investigations typify the reasoning and the advancement of story problem-solving modeling and resultant pedagogical considerations in effectively teaching the task. Comprehension or understanding is at the forefront of each investigation, as it should be. However, there are critical differences between each investigation about how understanding can be achieved by the problem-solver. These differences have served to inform the design of the MSASPT and are described as follows:

- Allen Newell and Herbert Simon - General Problem Solver (GPS) – were pioneers in cognitive science, approaching their studies from an artificial intelligence (AI) perspective. Their first efforts in modeling human problem-solving produced the *Logic Theorist* (LT). A computer program designed to use the elements of AI that they were developing to prove theorems in geometry (Newell, et al., 1958) with a high degree of accuracy demonstrating the potential power of artificial intelligence. Pamela McCorduck wrote that the LT was "proof positive that a machine could perform tasks heretofore considered intelligent, creative and uniquely human" (2004, p. 167). The GPS was developed using AI principles from their work with the LT and was designed to solve certain kinds of story problems.

Although both the LT and GPS's singular purpose was to solve simple story problems, not teach people how to solve story problems, simple extrapolation would suggest that both laid the foundation for intelligent tutoring system design and development. In 1970 an article by Newell and Simon appeared in Psychological Review, "Human Problem Solving: The State Of The Theory In 1970". This journal article seems to have facilitated Newell and Simon's seminal book, Human Problem Solving. With this book, they summarize and extend the ground-breaking work that they had done in artificial intelligence systems since the mid-1950's.

In this work they establish that there are two sets of thinking processes associated with the problem-solving process; understanding processes and search processes (Jonassen, 2010). These processes, operating within what Newell and Simon called the problem space, made up the pillars of their Problem Space Theory (PST). The important distinction in PST is that the search process is heuristic in nature (usually means-ends-analysis) and, as such, is not as dependent on deep structure understanding as other solution search methods (schema theory, ACT-R, etc.).

While heuristics such as means-ends-analysis can be effective in finding solutions to story problems, the downside is that knowledge structures are either incompletely formed or absent altogether, making the recall of an intact and useful solution schema virtually impossible and effectively preventing the transfer of problem-solving skills (Sweller, 1988; Hong, 1998). Certainly, Newell and Simon's model provides a rich framework for computational cognitive modeling to take place and a powerful source of cognitive science discovery, but it leaves story problem-solving methodology using schema and other learning theories wanting for lack of skill transfer.

- Walter Kintsch and Mitchell Nathan were groundbreaking cognitive scientists in the area of comprehension of text, distinctly different from Newell and Simon's AI perspective. They felt that story problem-solving was a reading intensive task in which poor text comprehension lead to serious errors in finding a solution (Nathan, et al., 1992). Their model for the human "understanding" process was called construction-integration (CI), and, after several years of study, they felt that it could be applied to the comprehension and solution of story problems. It evolved from studies of story comprehension and memory for text to a point where it could be considered as a possible architecture for that large area of cognition that is located between "perception" and "problem-solving" called understanding (Kintsch, 1988).

  Contained in the CI model is a definition of comprehension: "interaction and fusion between the to-be-comprehended object, usually a text, and the general knowledge and personal experience the comprehender brings to the situation" (Kintsch, 1988). Tying the comprehension process to the formation of knowledge structures or schemata further explains the concept. Donald Richgels explicates the whole problem-solving process around "understanding." Using Kintsch's theory, he believes that comprehension is the process of matching instances of the various story elements with an activated mental framework or schema. Put differently, meaning (comprehension) is not in the message itself, nor is it in the comprehender's schema in its abstract state; rather it is a result of a process that combines the two (Richgels, 1988).

Of course, it is important that there are descriptions of the comprehension process such as these, but a necessary correlation to a definition must be the methods that will be used to enable it to take place in the cognitive operations of a problem-solver. Working with story problems in the late 1970's and into the 80's, Nathan and Kintsch, along with the research of Kintsch and James Greeno, describe a problem-solving model that emphasizes not only the importance of reading for "understanding" (comprehension) but provides a method to use to gain understanding by way of the construction of what they call a *textbase*. The distinction here is that logically constructing a textbase provides for the deep structure comprehension of the problem, enabling knowledge structure construction, categorization of such structures, and subsequent transferability of skill from the categorical recall of pertinent knowledge structures.

- David Jonassen designed a problem-solving model that consists of sets (solution germane problem entities) used with other information (relationships, domain, etc.) to construct the situational and structural characteristics of the problem. The solution proceeds as this information is used to identify with an existing schema (or to establish the foundation for a new one), providing the necessary processing operations to solve the problem. The distinction here is that while Kintsch uses a list of propositions called a textbase, this model emphasizes the "understanding" of a story problem by giving the problem-solver a method whose steps are more fine-grained, but still intended to investigate its deeper structure for application to an appropriate resident schema(s) (Jonassen, 2003) facilitating skill transfer in the same way that Kintsch and his team did.

Kintsch et al. and Jonassen emphasized "understanding" the problem in their solution models in the belief that an inability to comprehend the problem text presents the greatest difficulty for problem-solvers. On the other hand, Neil Heffernan, in a 1997 study, found that the large effect of the composition factor (i.e., symbolization or algebraic representation of the quantities and relationships in a story problem), relative to the small or absent effect of comprehension tutoring hints in his experiments, provided a strong case against language comprehension as the major stumbling block for students. His research resulted in the design and development of the Ms. Lindquist intelligent tutoring system that addressed this symbolization difficulty. The hypotheses with which this paper and the design of the MSASPT are working with would dictate that both comprehension and composition performance are equally dependent on the ability to think abstractly and suffer equally from its absence.

In the Model Method, Swee Fong Ng and Kerry Lee also based their story problem-solving model on mitigating the difficulty that most students have with algebraically representing the quantities in a story problem. They use a heuristic to visually and concretely represent the quantities in the problem that are constrained by the relationships, both implicit and explicit, given in the problem text (Ng and Lee, 2009). Using a theoretical framework similar to Kintsch and Greeno (1985) they also incorporate common schemata such as, comparison, change, vary, etc. (Marshall, 1992), producing an effective story problem-solving model that combines the concretizing of an abstract process (algebraic representation) as co-incidental with the "understanding" process, much the same as Kintsch, Nathan, and Greeno did with their textbase generation.

The Model Method was very successful in teaching story problem-solving in Singapore elementary schools (Ng and Lee, 2009). In agreement with the hypothesis of limited conceptual thinking abilities stated earlier, I would suggest that the concretization of the representation process has a significant effect on this efficacy by easing the requirement for abstractive thinking. The MSASPT incorporates the concretizing of abstract problem-solving methods in ways similar to the Model Method, and they will be discussed in depth in Section IV.

Using a somewhat different approach, albeit while incorporating the formation of knowledge structures as well, Analogical Problem Solving (APS) is the use of a known solution to what is called a source problem to develop a solution to a novel target problem (Holyoak and Thagard, 1989). While APS has proven itself to be as effective as many of the other problem-solving methods, its use has not been incorporated in the MSASPT as a utility in the instruction process because of its complexity and the redundancies between the two methods. A similar problem-solving approach is *worked examples*. This mode of instruction, with its inherent concretization of the problem-solving process, is ideal for its inclusion in the MSASPT as a utility to enhance learning. Others would agree as it has been suggested that worked examples story problem-solving methods are very effective and easily adapted to intelligent tutoring systems (e.g., "Give Them Time to Think it Over") (Hilbert, et al., 2008). As such, they are a significant part of the MSASPT, at the step level, in various utilities that concretize processes, and through the entire problem-solving process. Instances of their use are described in Sections IV and VII (see Appendix P for more information).

Cognitive scientists opened the lid to the "black box" of the mind and produced many productive theories to explain and demonstrate how humans solve problems; in this case story problems. As a science inseparable from artificial intelligence and the computer, it wasn't long after Newell and Simon's work in building computational models (Logic Theorist and General Problem Solver) that, like B. F. Skinner before, people were building "teaching machines." This time, though, they had a computational power, only imagined in Skinner's days. And they were called Intelligent Tutoring Systems (ITS).

The purpose of reviewing ITS literature in this paper is to frame a discussion of the MSASPT, and for that reason, the review will be confined to the design and function of ITSs whose purpose is to provide one-to-one tutoring in some mathematical domain. There is a consensus that individual tutoring is the most effective form of educational interaction, at least for most domains. Bloom (1984) in his comparison of private tutoring with classroom instruction of cartography and probability found that 98% of the students with private tutors performed better than the average classroom student, even though all students spent the same amount of time learning the topics. John Anderson et al. (1985) recorded a four-to-one advantage for the private tutor, as measured by the amount of time for students to get to the same level of proficiency (Nwana, 1990).

While human tutoring seemed to offer some powerful evidence for its use, back in the day, research conducted by Kurt VanLehn in 2011, suggests that the efficacy of human tutoring doesn’t reach the commonly accepted effect size of 2.0. In fact, he found it to be less than 1.0 (0.79). The effect size of 0.76 for an ITS, effectively makes the argument that a quality ITS virtually matches the performance of a human tutor in facilitating greater learning gains than without either kind of tutor (VanLehn, 2011a). Factor in, as a result of both the educational reform movement and budget cuts to education resulting in larger class sizes, the idea of a portable, on-demand, one-to-one tutor for the nation's students is extremely compelling, especially in mathematics.

In 1985 there were over 10,000 pieces of educational software available (Anderson, et al., 1985), and surely there are much more now. Moreover, just as in 1985, most of them can be classified as Computer Assisted Instruction (CAI) in contrast to an ITS. Of course, this information begs the question, what are the differences between the two? In the interest of brevity and to better provide the qualifications for the MSASPT as an ITS (Section IV), the design and features that provide for the behavior of a computer program to function at a level approximately commensurate with a human tutor will be presented. While the behavior of all such artifacts is essentially the same, explicating the three different modes of how expertise is represented serves to define each type of ITS:

- Model tracing tutors (MTT) – expertise is presented using a problem solver created from a computational cognitive model of a particular problem-solving domain that provides algorithmically specific and detailed steps and that are arranged in precise yet flexible sequences. The steps contained in the expert model are used as comparators of the student steps; if a student step matches a step in the expert model, work continues- if not, the computer interacts with the student in some way to provide remediation for the error (VanLehn, 2011). The model tracing ITS is superior to the constraint-based tutor, described subsequently, with respect to the ability to provide accurate, contextual and even motivational remediation; this superiority increases with the complexity of the solution process goal structure, but it does take more time to develop a model tracing system (Kodaganallur, et al., 2005).

- Constraint-based tutors (CBT) – are based on the works of Ohlsson (1992, 1994) who advocated relying on a student's errors to build a student model as well as to provide remediation. Expertise is presented using comparisons to student input with built-in constraints on that input to determine student accuracy. In contrast to the model-tracing tutor, the connection to any diagnostic information the tutor might provide is not hidden in the sequence of student's actions, but in the problem state that the student is in (Mitrovic, et al., 2001). Constraints are crucial to the operation of a CBT with each constraint having a relevance state and a satisfaction state that specifies a condition that should hold for any correct solution satisfying the relevance state. When a student's work violates a constraint, specific information about the student's mental model is obtained and associated with a particular remediation observation designed to get the problem-solver back on a path that leads to a solution. The CBT does not consider essential to know how the student arrived at a particular problem state, only if there are constraints that have been violated. If no constraint violations occurred, the problem-solver continues on the chosen problem-solving path (Kodaganallur, et al., 2005).
- Example-tracing tutors (ETT) - in contrast to the first two tutors, use behavioral examples to provide expertise, without the use of machine learning/AI. They interpret a student's solution steps (and hint requests) with respect to a predefined solution graph for each problem contained in the tutor. The solution behavior graph is a directed, acyclic graph that represents acceptable ways of solving a problem. The links in the graph

represent problem-solving actions, and the nodes represent problem-solving states. A behavior graph may contain multiple paths, corresponding to different ways of solving a problem. It may also contain links that represent incorrect behavior, marked as such by the author who created the graph (Aleven, et al., 2009). The tutoring process consists of interpreting student actions for a specific problem against the behavioral graph for that problem (hence the term, *example-tracing*). Differences in responses trigger remediation designed to re-direct the student back to a correct solution, while correct responses qualify the problem-solver to proceed to the next step. The use of directed, acyclic graphs in ETT's is probably the most efficient method for recording possible problem-solving behaviors and their corresponding tutorial responses, but it is not the only way. In the discussion of the MSASPT design, another method will be described.

In spite of the differences enumerated above, all intelligent tutoring systems exhibit the same basic behaviors systematically designed into each tutor; the strategies, processes and monitoring capabilities that effective human tutors possess. In 2006, Kurt VanLehn essentially codified the behaviors of ITS's in his seminal work, "The Behavior of Tutoring Systems." This work serves as a non-technical introduction to ITS’s for the uninitiated and for those who know a great deal about ITSs, it serves as a proposed standardization of the descriptions of ITS behavior (VanLehn, 2006). Immediate feedback, step based, intelligent tutoring systems, of which the MSASPT and virtually all ITS's in use today, have two loops as shown in Figure 1 below:

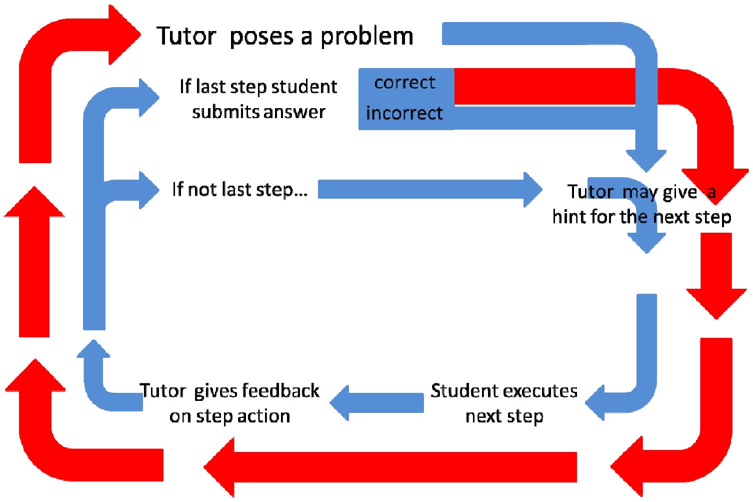

Tutor poses a problem
If last step student submits answer
correct
incorrect
If not last step...
Tutor may give a hint for the next step
Tutor gives feedback on step action
Student executes next step

Figure 1. The outer loop begins with the problem statement. The inner loop begins the problem-solving process continuing to cycle until the problem is answered correctly when control is transferred to the outer loop, and a new problem is given.

Presently, there are many instructional systems that have been engineered to exhibit these behaviors and accomplish the goal of effective tutoring; thus becoming classified as intelligent tutoring systems. Moreover, they operate in a wide variety of domains, furthering their status as integral to the goals of effective educational systems.

Section 4: The MSASPT Theoretical Framework:

The design of any instructional tool starts with a clear statement of the goal(s) to be achieved, proceeds to the pedagogy for achieving those goal(s), and ends with the means by which the instruction is to be presented. As hypothesized in Section II, an inability to deal with the mathematical domain of story problem-solving at a conceptual (abstract) level is thought to be the greatest barrier to story problem-solving success. The goal for the MSASPT is to sidestep that barrier and provide an operational method for success in that most Daedalean of enterprises; story problem-solving. Many of the other causes for difficulty can be dealt with by incorporating mitigation features into the ITS. In contrast, though, circumventing this barrier must be approached philosophically, epistemologically, and pedagogically as regards the general design of the ITS.

- Philosophically, credence must be given to the notion that, in most classrooms, teaching students how to solve story problems is vitally important, and worth the concerted effort necessary to do it. It must also be recognized that stunted abilities for conceptual thinking often occur in the domain of story problem-solving, but these same students are quite capable of such thought in other domains, such as reading, writing, and even science, as a result of their broad and successful experiences in these domains. The continued student futility in this endeavor gives credence to the root of the problem being with the instruction and, quite possibly with the instructor, either wittingly or unwittingly. Any story problem-solving pedagogy must be designed so that its principles and processes are knowable (understood) by both teacher and student, not just the student.
- Epistemologically, since knowledge acquisition involves the evaluation of evidence and inductive causal inference (Kuhn, et al., 1995), that evidence must be comprehensible to the student. Such evidence can be understood either concretely or conceptually, however, given the hypotheses of this study, evidence must be presented concretely.
- Pedagogically, the tutoring system must use methods of concretization as basic for the presentation of evidence; those relatively few students who are conceptually adept will advance quickly in the tutor and, quite possibly, have no need of it, but the conceptually disadvantaged will prosper.

The ascendant learning principle that has guided the design and development of the MSASPT tutor is drawn from schema-theoretic knowledge construction. Section V provides an in-depth explanation and rationale for a schema-based tutoring system. Briefly, schema theory holds that problem schemata are knowledge structures that are used to identify the type of problem being solved and that contain associated procedures for solving problems of that type (Blessing & Ross, 1996). Of course, these problem schemata must be constructed through the actual cognitive experiences of successful problem-solving; there is not a book that can be read or a pill ingested that will produce an intact story problem solution, without actually successfully solving the problem. "[P]roblem solving is not a uniform activity. Problems are not equivalent, either in content, form, or process. Schema-theoretic conceptions of problem-solving opened the door for different problem types by arguing that problem-solving skill is dependent on a schema for solving particular types of problems. If the learner possesses a complete schema for any problem type, then constructing the problem representation is simply a matter of mapping an existing problem schema onto a problem". (Jonassen, N/A).

It can be induced from a long history of ineffective story problem-solving that construction of story problem-solving schemata has remained largely unmanageable for the greatest majority of students engaged in a comprehensive study of mathematics. The MSASPT tutoring system design proceeds from the principled incorporation of learning theories, of which schema theory is the driving force. This force, combined with innovation borne of many years of teaching and investigation of story problem-solving, enabled the mitigation of difficulty in constructing problem-solving schemata and allowing successful story problem-solving to take place (see Appendix Q the design processes and features in MSASPT)

Section 5: The Role of Categorization in Solving Story Problems:

The concept of categories and the process of categorization are inseparable from schema theory, schemas, and schema building, upon whose principles MSASPT is based. However, before describing the synergistic relationship between categorization and schemas, an examination of the history and concepts of schema should be undertaken.

In 1911 Henry Head and Gordon Holmes postulated the notion of a "body schema." Such a schema consisted of anything that we are consciously aware of and is active in the movement of our bodies is added to a model of self (a schema) and becomes part of the individual schemas that make up the "body schema" (Bartlett, 1932). Twenty years later Frederick Bartlett expanded the scope of Head and Holmes' research to introduce the idea of schemas into cognitive psychology. Bartlett felt that a schema was the active organization of meaningful past experiences (not limited to physical movement). He also stressed the constructive character of remembering.

His famous "War of Ghosts" story demonstrated his notion of schema processing that when people try to recall a story, they reconstitute it in their terms, using existing, relevant schemas to shape their perceptions, rather than by rote memorization of details. The reconstituting process has the potential to initiate changes to existent knowledge structures producing new knowledge structures (schemas) (Bartlett, 1932). Unfortunately, Bartlett's theories of learning were not well received by the behaviorists, who held sway at the time, and his principles were discounted and lost the influence they merited had it been put forth at a later time.

In the early to mid-70's, with behaviorism on the wane and a much more effective approach to exploring and explaining human learning, Marvin Minsky, a pioneer in this cognitive science approach, was working to develop machines (computers) that were capable of intelligent behavior (artificial intelligence). During this time he came across Bartlett's work and subsequently reintroduced the notion of schema construction with his Frame System Theory. He postulated that when a new situation or problem is encountered a knowledge structure called a "frame" that has enough similarity to the present problem, is located in long-term memory and tailored to the new situation. Once this is done, the information attached to the frame (how to use the frame, what one can expect to happen next, what to do if these expectations are not confirmed, etc.) can be accessed (Minsky, 1975). Minsky's work is strongly suggestive of the structure and functions of what today's researchers would call schemas.

The essential role that categorization plays in a schema-based solution of algebra story problems was first demonstrated by Hinsley, Hayes, and Simon in a three-part experiment and published in a 1977 book chapter (Hinsley, et al., 1977):

- First, students were asked to classify over 70 different story problems from a high school algebra textbook by problem type. There were, on average, 13.5 categories containing more than one problem and considerable agreement as to category identity.
- Second, as a way to determine whether categorization was a function of student solution, problems were segmented and read to the student one segment at a time. After each segment, they were asked to categorize the problem. Half of the students were able to categorize the problem after hearing only one-fifth of it, supporting the assumption that students can categorize problems before having enough information to solve it.
- Third, six students who had worked on a complex distance, rate, time (DRT) problem had their verbal problem-solving procedures collected and examined. Three of the six categorized the problem as Pythagorean Theorem, while the other three recognized it as a DRT problem. The knowledge (schema) associated with each category determined what problem information was expected and what information was attended to, causing the first three students to consider irrelevant information and the second group of three to consider relevant information, suggesting that if the student had a schema for the correct category, the problem solution was facilitated by information retrieved from memory.

A summary of Hinsley and his research team's experimental results suggests that the role of categorization in story problem-solving is to enable the retrieval from memory of knowledge that the problem-solver has about a particular category of problem. The retrieved knowledge is in the form of a structure called a schema and includes useful equations, diagrams, and appropriate procedures for making judgments as to the relevance of the schema to the problem being solved (Hinsley, et al., 1977).

David Rumelhart, a noted cognitive scientist, further explicated the nature and role of schemas as a theory about knowledge - about how knowledge is represented (structured) and about how that representation facilitates the use of the knowledge in productive ways. This theory postulates that all knowledge is packaged into units called schemas (or schemata). Contained in these packets of knowledge is not only the knowledge itself but information about how this knowledge is to be used. (Rumelhart, 1991). Rumelhart (1991) believed that schemas are a somewhat informal, private, implicit theory about the nature of the events, objects, or situations we experience. The total schemata set drawn from our experiences in trying to understand the world constitutes a personal theory of the nature of reality. The total set of schemata in use at a particular moment represents an internal model of the situation at hand. As his theory regards story problem solving, a schema includes a categorical reference and associated knowledge for that reference (Rumelhart, 1978).

Later, Sandra Marshall (1992) characterized schemas as patterns of relationships as well as their connection to operations and detailed the four types of associated knowledge contained in a comprehensive story problem-solving schema:

- Constraint or schema knowledge has to do with pattern recognition. An informational pattern in a long-term memory resident schema is compared to the current story problem;
- Feature or elaboration knowledge has to do with deciding whether the necessary elements are present in the problem, given that the pattern has been recognized as characteristic of a certain schema. Several potential patterns within one schema may exist in a problem, and the most reasonable or most likely one for a solution needs to be recognized.
- Planning knowledge enables the drawing of inferences, making estimates, creating goals, and developing plans using the framework from the first two;

- Execution knowledge utilizes skills, procedures, or rules as needed when faced with a problem for which this particular framework is relevant.

She also suggested that the role of a particular schema (knowledge structure) created in long-term memory is to organize information from similar problem-solving experiences. The schema can then enable the problem-solver to differentiate new experiences (problems) as similar or dissimilar. Similarity facilitates the application of production rules and other processes contained in the schema. Dissimilarity promotes the rudimentary beginnings of a new schema or the adjustment of an old one to fit new information. Her explanation of the schema-based problem-solving process as one that occurs over four stages follows:

1) The identification or recognition of applicable problem schemas using long-term memory resident schema knowledge.
2) The representation of the problem. Mapping key elements of the problem type's schema and their relationships to the given problem.
3) The planning and selecting of an appropriate operation (e. g., multiplication, division) or setting up a mathematical solution equation.
4) The last stage is to carry out the plan.

(Marshall, 1995)

Conceptual knowledge in mathematics is a network of discrete pieces of information linked to understanding the relationships that unite them into a unified concept.

- The relationship that exists between the base and height of a triangle
- Visualizing a representation that accurately exhibits the relationship between two quantities in the problem as one being three times more than the other

A learner has conceptual knowledge in mathematics only if these relationships are recognized and understood (e.g., the first example's derivation from a parallelogram) (Hiebert & Lefevre, 1986). As independent or new pieces of information are organized and related to one, another conceptual growth occurs (Lawler, 1981). Piaget named this process accommodation. Procedural knowledge has two components:

- The formal language of mathematics (symbols and syntax)
  - The difference between a constant and a variable
  - The difference in result between (3 – 5)/3 and 3 – 5/3
- The rules, algorithms, and processes needed to accomplish mathematical tasks.
  - Finding the representational value of a specified unknown in an equation that has several unknowns.

A learner has procedural knowledge in mathematics if these skills and abilities are apparent in solitary mathematical efforts (Post & Kramer, 1989). Figure 2 is a flowchart that illustrates the schema-based problem-solving process adapted from Xin (2008).

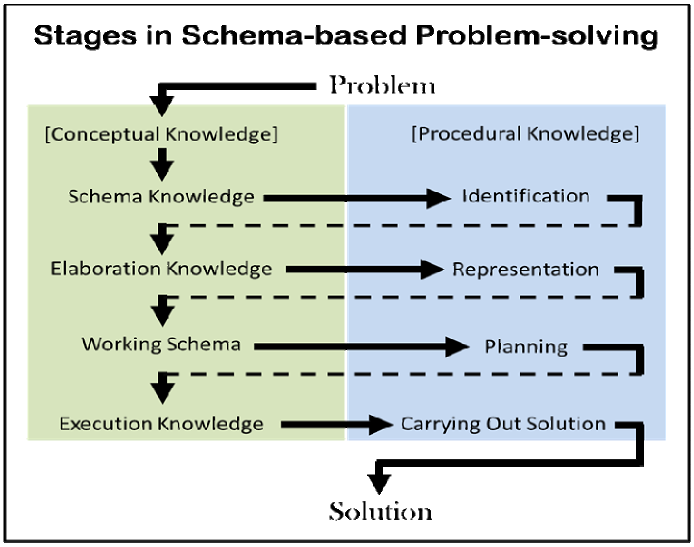


Figure 2. Conceptual knowledge (if present) working with the resident collection of knowledge schemas, facilitates the identification of a pertinent schema. Elaboration of the chosen schema fills in missing information to build a working schema. Solution planning based on the working schema takes place, producing executable solution equation and the solution to the problem.

Taking care to point out the obvious, schema-based learning involves a considerable amount of abstractive (conceptual) thinking. However, just as we can learn the relationship between the areas of a rectangle and a triangle, and access it when needed, MSASPT converts the concepts involved in story problem-solving to concrete processes even as practice brings them within the problem solver's domain of conceptual knowledge. Ready for use in problem-solving experiences they produce increasingly effective outcomes from the expanding depth and breadth of a student's problem schema. This is what ultimately determines the accuracy of his/her problem representation and solution success (Chi, et al., 1981).

Many, if not virtually all, of the educational psychologists, cognitive scientists, and mathematical pedagogists that have drawn distinctions between expert and novice story problem-solvers have stated that experts have an extensive quantity of comprehensive domain-specific schemas at their disposal, acquired through problem-solving experiences. Novices have far fewer of these schemas, which may well be incomplete. They must then resort to domain-independent heuristics to solve the immediate problem. Unfortunately, this provides no path for transfer of story problem-solving skills to other types of story problems (Jonassen, 2000; Jonassen, 2003; Hegarty, et al., 1995; Sweller, 1988; Chi, et al., 1981; VanLehn, 1989; Nathan, et al., 1992). The importance of schema generation, singularly provided by varied problem-solving experiences cannot be overstated.

It follows as well that the process of problem categorization and its essential role in identifying the correct schema in long-term memory provides the problem-solver with a script for a solution. Without its identifying characteristics not only would the solution script be unavailable for the present problem, but the subsequent transfer of problem-solving skill to dissimilar problems would be very difficult, consuming time and short-term memory stores, for the problem-solver to remark on the congruency between a held schema and the problem to be solved. Section VI identifies and discusses the methods that have been employed by story problem-solving pedagogists over the history of efforts in this domain.

Section 6: Review of Categorization Theories and Methods:

With the synergistic role of problem categorization in a schema theoretical approach to story problem-solving firmly established, it is imperative that the learner is provided with a categorization process that is accurate, discriminating, and economical. Eleanor Rosch, a widely cited authority on categorization, proposes two general categorization principles and how these principles might be used for the formation of categories:

- Cognitive Economy has to do with the function of category systems and asserts that the responsibility of category systems is to provide maximum information with the least cognitive effort;
- Perceived World Structure has to do with the requirement that the arrangement of the categorical information comes as structured information that reflects the real world as perceived by the learner rather than as arbitrary or unpredictable characterizations.
- These principles are achieved if the generated categories map the perceived world-structure as closely as possible.
    - This can be accomplished in two ways
        - by the aligning of categories to given structural characteristics
        - by the definition or redefinition of characteristics to produce a given category that is appropriately structured

(Rosch, 1978).

Edward Smith and Steven Sloman proposed that two distinct processes can be used to categorize.

- A rule-based process that is relevant to theory-driven categorization is deliberative, analytic, and capable of providing explicit justification for a categorization decision.

- A similarity or match-based process that is more automatic and holistic, and it cannot be used to supply convincing justifications for categorization decisions.

Additionally, they felt that the process of categorization was like language and reasoning in that it often reflects the operation of both similarity and rules (Smith & Sloman, 1994). Later, in a journal article, Smith, Andrea Patalano and John Jonides added two more processes that could be used for categorization:

- the similarity of the test object to a prototype of the category
- whether the features of the test object are best explained by the 'theory' that underlies the category (Smith, et al., 1998, p. 169)

With regards to story problem-solving, rule-based and similarity-based categorization processes have the greatest association with story problems and will be referred to, along with Rosch's principles, in the remainder of this section and again in Section VII, regarding the MSASPT categorization process. It is important to note that in undirected categorization tasks, students first attempt to use rule-application and generate a simple rule or even several. When this proves too daunting or even impossible, the student switches to exemplar similarity as in their rule search they have inadvertently come to memorize the exemplars and their associated category labels. Eventually, the exemplar-similarity procedure is capable of producing correct categorizations and becomes the preferred method. Even when a categorization rule is present, students will use the exemplar-similarity procedure if the test object contains information that is characteristic of one of the target categories. (Smith, et al., 1998).

Rule-application and exemplar-similarity categorization procedures obviously differ in the sources of information for category determination, but they also differ in the extent to which they involve, respectively: "(a) analytic [serialistic] vs. holistic processing, (b) differential vs. equal weighting of attributes, (c) instantiation of abstract conditions vs. matching concrete information, (d) high vs. low loads on working memory, (e) serial vs. parallel processing, and (f) strategic vs. automatic processing" (Smith et al., 1998, p. 170). Several of these dichotomies have implications for the

MSASPT and its attention to first the circumvention, then the remediation of the insufficiency that mathematics students exhibit for abstract thought and successful story problem-solving. These implications will be considered in detail in Section VII.

Historically, the undeniably critical role of problem categorization has been done quasi-implicitly, which is to say, from an understanding of the problem text. Many educational psychologists, cognitive scientists, and mathematics pedagogists advocate for the use of the problem's structural content (its solution equation), to categorize story problems (e.g., Sweller, 1988; Holyoak, 1995; Jonassen, 2003; Hinsley, et al., 1977). For example:

- If one angle in a scalene triangle is 60° and a second angle measures 88°, what must be the measure of the third angle?

A reading of the text yields the deep structure of this problem as the solution equation derived from the geometric axiom that all angles in a triangle must total 180° $\therefore 60° + 88° + y° = 180°$; $148° + y° = 180°$; $y° = 180° - 148°$; $y° = 32°$.

Other researchers (e.g., Chi, et al., 1981; Blessing & Ross, 1996; Mayer, 1981) advocate the use of both the situational (surface features) and structural content of the problem creating a synergistic relationship that leads to a quicker identification of the problem's solution schema, limiting the scope of a search to only those concerned with, in this case, triangles. Using the situational characteristics alone may cause a miscategorization and an erroneous solution (e.g., use of the Pythagorean Theorem as the solution equation because it is associated with triangles). For problem-solvers whose focus might be only on the structural characteristics of a problem, answers that are situationally impossible are sometimes produced (Hinsley, et al., 1977).

Physics story problem solution lends itself well to this quasi-implicit categorization because there is always an underlying principle connected to a solution equation as part of the structure of a correctly selected schema. Story problems in other domains, such as mathematics, are often situated in contexts that have apparent, even obvious, deep structures (solution equations) associated with them. Geometry, trigonometry, ratio and proportion, simple interest, simple motion (DRT), uniform motion ($D_1 = R_1T_1$ and $D_2 = R_2T_2$), etc., lend themselves well to quasi-implicit categorization. However, there remains a certain significant class of story problems in mathematics that do not have readily apparent deep structures, and, in fact, the relationships (implicit and explicit) that may exist in such problems must be investigated, many times with a great deal of acumen, to produce a solution equation.

The solution of mental manipulatives (Novotna, 2001), as these kinds of story problems are often called, is dependent entirely on the problem-solver's ability to synthesize her own solution equation. Age and money problems are prime examples of mental manipulatives. For example: Presently, Bill is 5 years more than three times as old as his nephew, John. In 10 years, he will be twice as old as his nephew, John. How old is Bill now? Very different from story problems with a discoverable, pre-existing solution equation, they require the use of an explicit method of categorization.

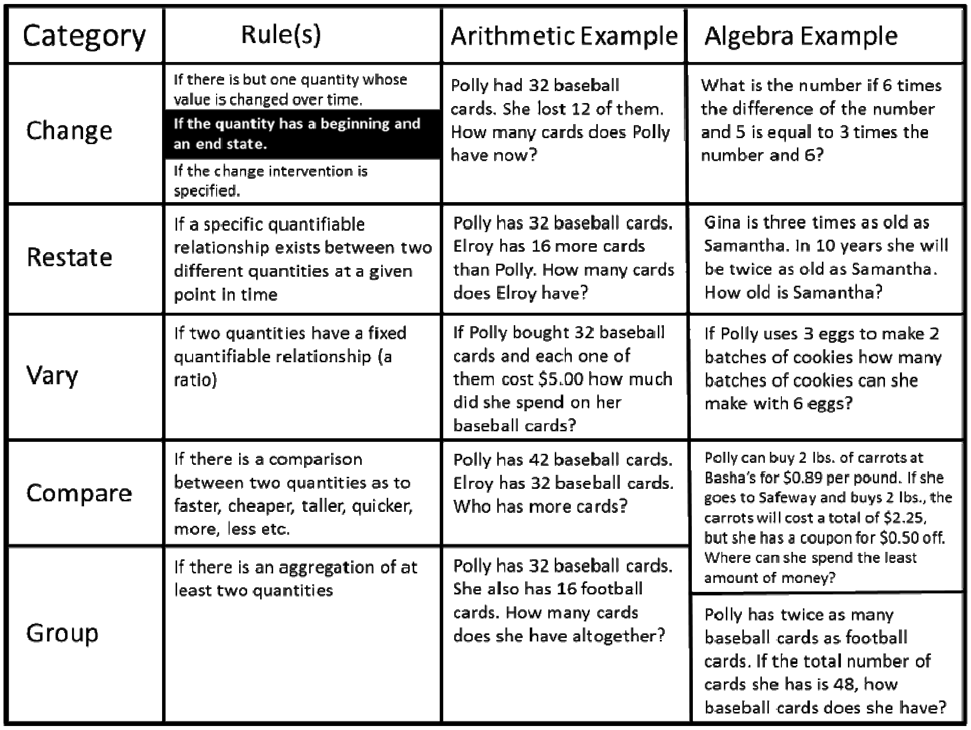
Other researchers in the domain of story problems such as Sandra Marshall, Edward Smith and his team, and Richard Mayer, along with Eleanor Rosch, developed methods to categorize story problems explicitly. Explicit categorization of story problems may utilize:

- The similarity to an exemplar
- Rules (usually in a multi-step process)
- A descriptive term with associated properties and features
- Rosch's methods of real-world perception to facilitate the formation of a category of story problems.

In using any of these explicit methods of categorization, care must be taken to include discriminatory classifications for problem differentiation, but not so discriminatory that the number of categories exceeds the economic limits of efficiency for the enterprise or by differentiation that is irrelevant to the purpose of solving story problems (Rosch, 1978).

For mental manipulatives problem categorization, similarity to an exemplar could function at the situational level where superficial aspects of the problem exist and can be compared to the same aspects of a target problem. However, at the structural level, where no deep structure is immediately apparent, similarity to an exemplar deep structure (solution equation) is largely unavailable, except in the simplest of these kinds of problems. Just as using only situational content in quasi-implicit categorizations, using exemplar similarity to find a matching schema, may lead to errors in mental manipulatives solutions. Schema choice based on similarity can reduce the solution search, but extreme care must be taken to ensure that the resultant solution equation is correct as no solution equation is immediately apparent to serve as a cross-check.

Many explicit categorization processes are rule-based but rely on operations-based production rules. On the other hand, Sandra Marshall 's system is situation-based using simple production rules (Yeap & Kaur, 2001). Used widely in other story problem-solving research, pedagogies, and intelligent tutoring systems (e.g., Xin, 2008; Yeap & Kaur, 2001; Derry, et al., 2001; Yolles, 2006), these five categories exist as schemas in the domain of arithmetic and algebra story problem solving as shown in Figure 3 below.



Category	Rule(s)	Arithmetic Example	Algebra Example
Change	If there is but one quantity whose value is changed over time. If the quantity has a beginning and an end state. If the change intervention is specified.	Polly had 32 baseball cards. She lost 12 of them. How many cards does Polly have now?	What is the number if 6 times the difference of the number and 5 is equal to 3 times the number and 6?
Restate	If a specific quantifiable relationship exists between two different quantities at a given point in time	Polly has 32 baseball cards. Elroy has 16 more cards than Polly. How many cards does Elroy have?	Gina is three times as old as Samantha. In 10 years she will be twice as old as Samantha. How old is Samantha?
Vary	If two quantities have a fixed quantifiable relationship (a ratio)	If Polly bought 32 baseball cards and each one of them cost $5.00 how much did she spend on her baseball cards?	If Polly uses 3 eggs to make 2 batches of cookies how many batches of cookies can she make with 6 eggs?
Compare	If there is a comparison between two quantities as to faster, cheaper, taller, quicker, more, less etc.	Polly has 42 baseball cards. Elroy has 32 baseball cards. Who has more cards?	Polly can buy 2 lbs. of carrots at Basha's for $0.89 per pound. If she goes to Safeway and buys 2 lbs., the carrots will cost a total of $2.25, but she has a coupon for $0.50 off. Where can she spend the least amount of money?
Group	If there is an aggregation of at least two quantities	Polly has 32 baseball cards. She also has 16 football cards. How many cards does she have altogether?	Polly has twice as many baseball cards as football cards. If the total number of cards she has is 48, how baseball cards does she have?

Figure 3. Marshall's taxonomy is an excellent example of explicit categorization, but there are few examples to be had, otherwise.

Another source of information regarding explicit categorization is Richard Mayer's analysis of over 1000 story problems from ten high school and college mathematics textbooks, which includes an abundance of categorization principles, methods, and examples (Mayer, 1982). His work is still judged by many scholars as the most comprehensive categorization of mathematics story problems (Jonassen, 2003). However, with most of his categorizing taking place quasi-implicitly, via structural and situational content, the work doesn't advise the question of how mental manipulative story problems are categorized, except for a small but very significant broad distinction of problem type.

The problem type distinction that he makes coincides precisely with the distinction made in this paper between quasi-implicit categorization and explicit categorization. At the top-most level in Mayer's taxonomy of problem types is whether a problem is *source coded* or *non-source coded* (Mayer, 1982). A *source coded* problem is one that has a solution equation associated with it and as such can be quasi-implicitly categorized for schema construction and retrieval according to both its solution equation (deep structure) and its situational content. For example:

- Kendra can ride a bike at 12 miles per hour (mph). Disregarding acceleration to this speed, how long will it take to get to the hardware store 4 miles away in minutes?

This problem has a solution equation associated with it (D = RT) as determined by the quantities in the text (rate, time, and distance) so the source of its solution is a pre-existing equation that need not be generated by the problem-solver. Additionally, riding a bike is the kind of situational content that will point a reasonably experienced problem-solver to the possibility of a DRT solution equation upon reading that part of the problem text the first time. On the other hand, a *non-source coded* problem is one that does not have a solution equation associated with it and must be categorized by using an explicit process to facilitate schema construction and retrieval. For example:

- Vijay has a total of 17 coins that total $2.85. If he has only nickels and quarters, how many quarters does he have?

This problem does not have a solution equation associated with it as no pre-existing equation involves nickels and quarters, requiring the problem-solver herself to generate the equation. It may be argued that the situational content involving coins, for an experienced problem-solver, will possess similar schemas containing solution procedures in memory. However, the neophyte problem-solver, with little experience, will draw a blank with no structural content and no situationally similar schemas as yet in memory. Moreover, situational similarity for an experienced problem-solver leads to a narrower search, but may not produce the correct answer.

In a rather Catch-22 situation, how is a *non-source coded* problem to be categorized when a consistent, unchanging solution equation is absent? And if there is no apparent deep structure at the beginning of the solution process, must one use the rather mercurial (especially for a beginner) situational structure alone to retrieve the solution schema? In Section VII the answer to this conundrum is presented, along with its theoretical framework and its integration into the MSASPT Categorization step.

Section 7: The Theoretical Framework, Processes, and Features for Learning Problem Categorization in the MSASPT:

This section considers how a theoretical framework, drawn from the concepts, principles, and processes explicated in Section VI and targeted at the hypothesized conceptual thinking difficulty that seems to be almost omnipresent in story problem-solvers, has provided for the design of the MSASPT problem categorization process. Before such a process can be elaborated the responses to two fundamental questions must be ascertained:

- On the question of whether the categorization process should be quasi-implicit or explicit
    - The GED Mathematics test includes both mental manipulatives *(i.e., non-source coded)* and *source coded* problems from the domains of mathematics and physics
    - Question response: categorization should be *explicit* to allow for the absence of a solution equation that pre-dates the problem and could have been used for deep-structure problem identification.
- On the question of whether the categorization process should be rules-based or similarity-based
    - Rules-based categorization schemes are germane to theory-driven categorization (Smith & Sloman, 1994)
    - Rules-based categorization schemes are deliberative, analytic, and capable of providing explicit justification for a categorization decision (Smith & Sloman, 1994) which is an exceedingly important feature in an intelligent tutoring system
    - A similarity or match-based process is counter-indicated by its largely holistic (abstract) nature. Moreover, it cannot be used to supply convincing justifications

for categorization decisions (Smith & Sloman, 1994). Such an absence would negatively impact an intelligent tutoring system's effectiveness.

- Rules-based categorization (step-by-step) is situated much more at ease with the serialistic learners that make up the great majority of students who struggle with story problems (Kozhevnikov, M., 2007).
- Question response: categorization should be rules-based.

Based on the responses produced vis a vis the two fundamental questions above, the MSASPT categorization process is explicitly presented from a rules-based platform.

According to Rosch (1978), cognitive economy and a student perceived world structure are the two foundational principles of categorization. Cognitive economy is achieved by a process designed to provide maximum information with the least cognitive effort and perceived world structure is categorical information that reflects the real world as perceived by the learner rather than arbitrary characterizations. She offers two means for this to take place: (1) by aligning categories to given structural characteristics and (2) by the definition or redefinition of characteristics to produce a given category that is appropriately structured.

In the context of Rosch's two categorization principles, cognitive economy is achieved by the identification of the critical elements common to the domain of story problems found on the GED Mathematics test and in Algebra I textbooks. It is proposed that there are five of these elements and that these five serve as characteristics that make up the category in which the story problem resides. The term "characteristic" has been carefully chosen to indicate a more integrated category and one that with experience becomes implicitly known to the problem-solver. Moreover, with experience, the first three characteristics take an active role in the characterization process, while the last two become re-active according to the first three; characteristic rather than classification is a more flexible term in that situation.

The means by which a perceived world structure is achieved is by a rules-based definition for each of a story problem's critical elements (characteristics). The category remains an amalgam of these characteristics and, as noted above, is not singularly named as Marshall's categories are. The five elements or characteristics in the MSASPT categorization process have dichotomous rule-based definitions except for one (the second), which is trichotomous. The name of the characteristic and its defining rules are as follows:

1) Quantity Number
    a. If the story problem (SP) has a single quantity whose value changes over time in some (mathematically) specified manner, it is classified as a *change quantity* problem.
        i. This is the same condition set forth for Marshall's "change" problems.
        ii. Example: George has twice as many marbles as he had two days ago. If he had "y" marbles then, write an expression for how many marbles he has now.
    b. If the SP names at least two quantities that are germane to its solution, it is classified as a *multiple quantity* problem.
        i. Example: George has twice as many marbles as Henry. If Henry has "y" marbles, write an expression for how many marbles George has.
2) Quantity Relationships
    a. If any pair of quantities in the SP are shown to be dependent on one another for their value, it is classified as a *relational* problem. Only paired relationships are considered.
        i. Example: Kaila worked two hours longer than Dena did. If Kaila worked "y" hours, write an expression for how many hours Dena worked.
    b. If at least two quantities are given in terms of their total, it is classified as a *related-by-total* problem.

        i. Example: Together, Kaila and Dena have a total of 25 nickels. If Kaila has "y" nickels, write an expression for how many nickels Dena has.

   c. If there are no existent relational pairs of quantities related by their total in the SP, it is classified as a *non-relational* problem.

        i. Example: Dena rented a surfboard at the beach. She paid a flat fee of $5.00 and agreed to pay "$y" for each hour she had the board. If she had the board for 2 hours, write an expression that represents how much she paid altogether.

3) Solution Equation Source

   a. If the SP cannot be associated with any pre-existing solution equation by its named quantity's congruency with the quantities present in a pre-existing solution equation, it is classified as an *internally coded* problem.

        i. Internal, as solution code (equation) coming from within the problem text.

        ii. This characteristic is identical to Mayer's *non-source coded* SP classification.

        iii. These kinds of problems are known variously as mental manipulatives, number or puzzle problems.

        iv. Example: Kaila and Dena have 76 Facebook friends between them. If Kaila has 12 more friends than Dena, how many friends does Kaila have?

   b. If the SP can be associated with any pre-existing solution equation by its named quantity's congruency with the quantities present in a pre-existing solution equation, it is classified as an *externally coded* problem.

        i. External, as solution code (equation) coming from outside the problem text.

        ii. This characteristic is identical to Mayer's *source coded* SP classification.

iii. Mayer’s SP categorization (1982) taxonomy consisted almost entirely of differentiated *source coded* problems. To achieve cognitive economy the MSASPT categorization architecture, instead, treats his very large number of categories as monolithic with the idea that the domain knowledge required for any of these categories can be selectively provided by the human instructor overseeing the learner according to the goals of the particular classroom instruction.

iv. Example: A plane traveled 450 miles at two different speeds. The first part of the trip was flown at an average speed of 105 mph. The second part of the trip was flown at an average speed of 115 mph. If the trip took a total of 4 hours, for how long did the plane travel at 115 mph? (complex DRT)

4) Solution Equation

a. If the SP has been characterized as *internally coded,* then the solution equation is to be generated by completing subsequent problem-solving steps.

i. Therefore “none yet” is entered as the fourth characteristic.

b. If the SP has been characterized as *externally coded,* then the solution equation is identified and entered as the fourth characteristic.

i. $D_1 + D_2 = D_{total}$ and $R_1T_1 + R_2D_2 = 450$ mi.

ii. It is important to note that all externally sourced solution codes used in a particular MSASPT domain application are accessible to the problem-solver through the use of an instructor-customizable utility, just as solution equations are accessible to the GED Mathematics test taker.

5) Solution Type

a. If the SP text requires the answer to be in the form of an expression, the characteristic is *expression*.

        i. Typically, if the problem text does not end with a question mark, the solution type characteristic is *expression*.

        ii. Example: Together, Kaila and Dena have a total of 25 nickels. If Kaila has "y" nickels, write an expression for how many nickels Dena has.

    b. If the SP text requires the answer to be in the form of a number, the characteristic is *numerical*.

        i. Typically, if the problem text ends with a question mark, the solution type characteristic is *numerical*.

        ii. Example: Kaila and Dena have 76 Facebook friends between them. If Kaila has 12 more friends than Dena, how many friends does Kaila have?

Taken on its face, four dichotomous characteristics and one trichotomous characteristic would be a combinatorial problem that yields 48 (2 • 3 • 2 • 2 • 2) discrete characteristic collections or problem categories. However, reductions in this number emerge with the following considerations:

- If, in the first characteristic, *change quantity* is chosen
    - The second characteristic must be *relational* as a pair of quantities (original and new) are dependent on one another for their values, making this characteristic unitary.
    - The third characteristic must be *internally coded* as no pre-existing solution equation is associated with change quantity problems, making this characteristic unitary.
    - The fourth characteristic must be "no solution code yet" as the SP is *internally coded* and its solution equation is to be developed over the next few MSASPT steps, making the characteristic unitary.
    - The fifth characteristic can be either *expression* or *numerical* making it dichotomous.

    - The combinatorial result is 2; (1 • 1 • 1 • 1 • 2).
- If, in the first characteristic, *multiple quantity* is chosen
    - The second characteristic can_be any of the three by its rule.
    - The third characteristic can be either by its rule.
    - The fourth characteristic must be "none yet" if the SP is *internally coded*. If the SP is *externally coded* the solution code must be provided. This effectively makes the fourth characteristic unitary because it is entirely dependent on the third.
    - The fifth characteristic can be either *expression* or *numerical*.
    - The combinatorial result is 12; (1 • 3 • 2 • 1 • 2).

The number of SP categories, then, is 14 (2 + 12). While the fifth characteristic (solution type) is much less nuanced than the other four, it is included in the categorization process to ensure that the problem-solver recognizes how the answer to the problem must be presented. Many times, both on the GED Mathematics test and in Algebra I textbooks, answers are to be presented as *expressions* and finding a *numerical* answer would either be a waste of test time or impossible without more information. Graphical representations of the above combinatorial derivations can be found in Appendix R.

Remembering that the hypothetical cause of story problem-solving difficulty is an inability to think conceptually (i.e., abstractly or to make generalizations), each design decision made for the MSASPT must first address that problem, either directly or indirectly. In Section II it was suggested that a contributor to an inability to think abstractly was a learner's serialistic cognitive style in which s/he considers information most efficiently if provided sequentially and in small bits and pieces. Therefore, the MSASPT categorization process is conducted by determining the proper characteristic for each of the five elements always in the same sequence. Using this coding process, analysis of information proceeds in successive steps so that each step provides cues for the processing of later steps. This is in opposition to simultaneous (parallel) coding processes that are used when all the pieces of information or all the stimuli are surveyable at one time and are thus available for processing at one time (Hickman, n/d).

Paul Wachtel wrote, “We are always constructing reality every bit as much as we are perceiving it” (1980). I would suggest that his statement, succinct and accurate, describes schema theory (and other learning theories) precisely. The intelligent tutoring system named Ms. Stephens Algebra Story Problem-solving Tutor (MSASPT) consists of six major steps, which includes the solution of the solution equation. All but the last step (solving a first or simple second-degree algebraic equation) have detailed practice programs associated with them. The practice program designed and developed for the Categorization Step is a self-contained ITS to provide categorization instruction and practice in the same way that the MSASPT does for the entire story problem-solving experience. This paper will now turn its attention to the MSASPT Categorization Practice Utility (MS-CPU) process that enables the perception and construction of story problem reality.

The MSASPT Categorization Process:

The categorization process is illustrated in Figure 4 as a flowchart below.

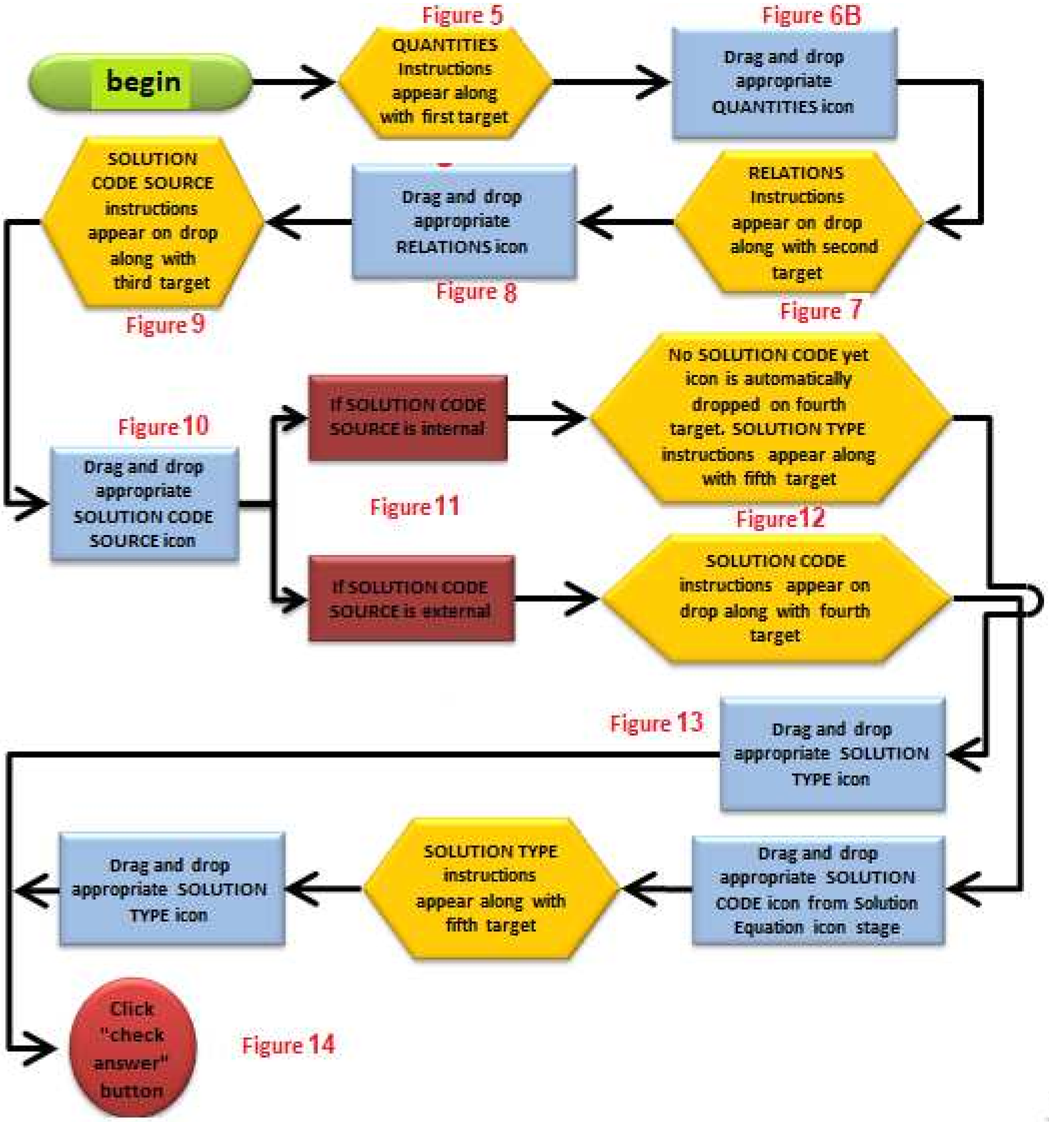
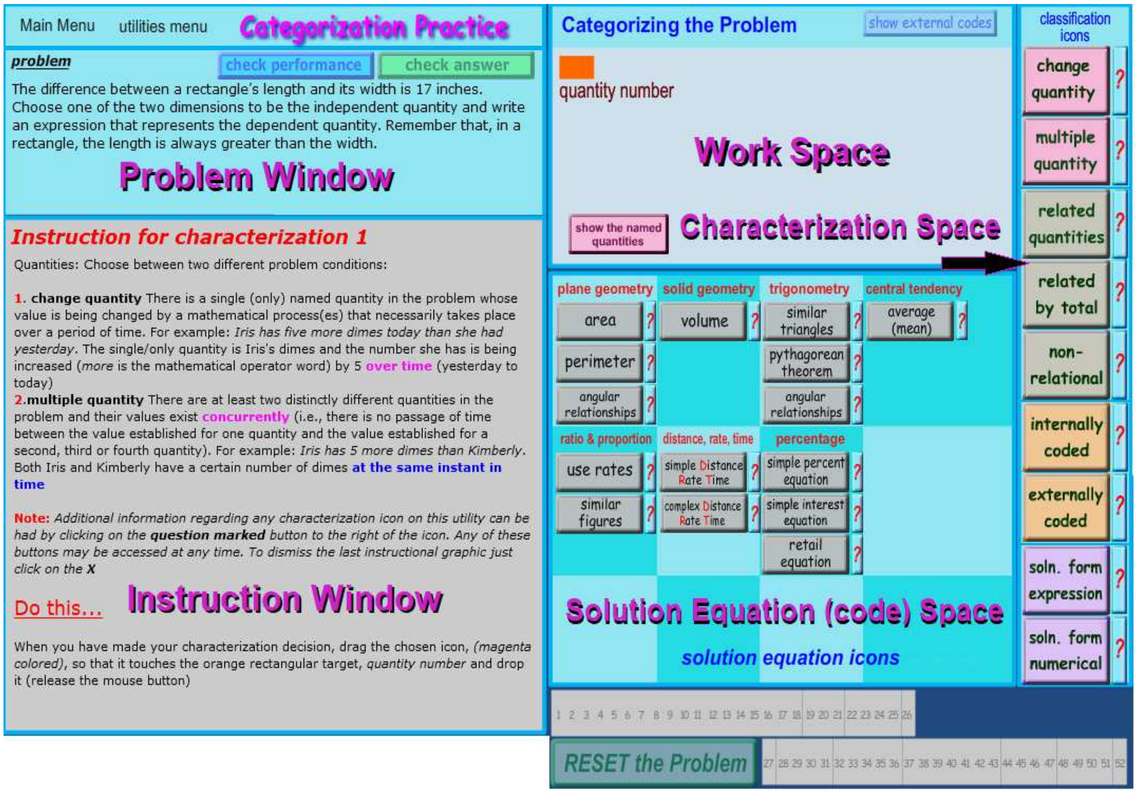

begin
Figure 5
QUANTITIES instructions appear along with first target
Figure 6B
Drag and drop appropriate QUANTITIES icon
RELATIONS instructions appear on drop along with second target
Figure 7
Drag and drop appropriate RELATIONS icon
Figure 8
SOLUTION CODE SOURCE instructions appear on drop along with third target
Figure 9
Figure 10
Drag and drop appropriate SOLUTION CODE SOURCE icon
If SOLUTION CODE SOURCE is internal
Figure 11
If SOLUTION CODE SOURCE is external
No SOLUTION CODE yet icon is automatically dropped on fourth target. SOLUTION TYPE instructions appear along with fifth target
Figure12
SOLUTION CODE instructions appear on drop along with fourth target
Figure 13
Drag and drop appropriate SOLUTION TYPE icon
Drag and drop appropriate SOLUTION CODE icon from Solution Equation icon stage
SOLUTION TYPE instructions appear along with fifth target
Drag and drop appropriate SOLUTION TYPE icon
Click "check answer" button
Figure 14

Figure 4. Each interval on the chart is keyed to an illustration (screenshot) following its description for the next ten figures. The descriptions and illustrations are presented sequentially and show the changes in the user interface as the student moves through the process.

The user interface (UI) as it appears at the beginning of the problem categorization session is shown in Figure 5.



The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quantity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

Instruction for characterization 1

Quantities: Choose between two different problem conditions:

1. **change quantity** There is a single (only) named quantity in the problem whose value is being changed by a mathematical process(es) that necessarily takes place over a period of time. For example: *Iris has five more dimes today than she had yesterday*. The single/only quantity is Iris's dimes and the number she has is being increased (*more* is the mathematical operator word) by 5 **over time** (yesterday to today)

2.**multiple quantity** There are at least two distinctly different quantities in the problem and their values exist **concurrently** (i.e., there is no passage of time between the value established for one quantity and the value established for a second, third or fourth quantity). For example: *Iris has 5 more dimes than Kimberly*. Both Iris and Kimberly have a certain number of dimes **at the same instant in time**

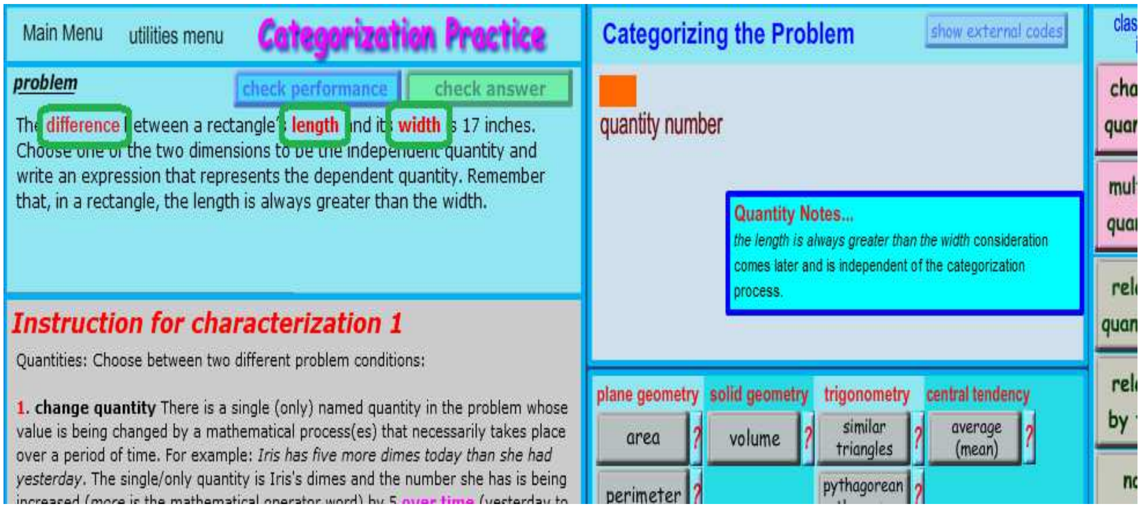
Note: *Additional information regarding any characterization icon on this utility can be had by clicking on the* ***question marked*** *button to the right of the icon. Any of these buttons may be accessed at any time. To dismiss the last instructional graphic just click on the* ***X***

Do this...

When you have made your characterization decision, drag the chosen icon, *(magenta colored)*, so that it touches the orange rectangular target, *quantity number* and drop it (release the mouse button)

Figure 5. The various UI windows, spaces, and stages are identified in red, but the identifications themselves do not appear on the actual UI. The **begin** button has been clicked (and is out of sight), a problem appears in the Problem Window and instructions for the first categorization step appear in the Information and Instructions Window. The first drag and drop (D-D) target (orange rectangle) with the characterization label (*quantity number*) appears in the Work Space. Only the two permissible icon choices are D-D enabled.

The **show the named quantities** button has been clicked (and disappeared) to produce figure 6A.



problem

The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quantity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

Quantity Notes...

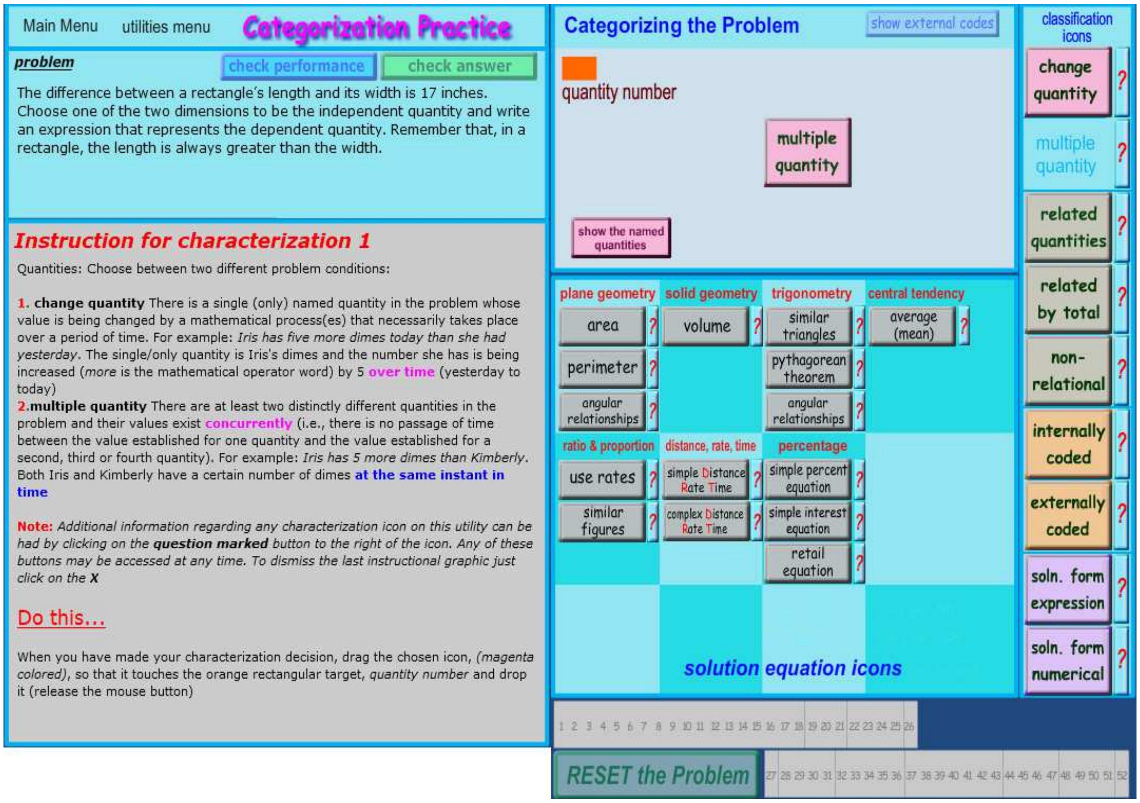
the length is always greater than the width consideration comes later and is independent of the categorization process.

Instruction for characterization 1

Quantities: Choose between two different problem conditions:

1. **change quantity** There is a single (only) named quantity in the problem whose value is being changed by a mathematical process(es) that necessarily takes place over a period of time. For example: *Iris has five more dimes today than she had yesterday*. The single/only quantity is Iris's dimes and the number she has is being increased (*more* is the mathematical operator word) by 5 **over time** (yesterday to

Figure 6A. For additional help, this action highlights in red, the quantities germane to the solution of the problem, length and width, as shown. After a certain number of uses, this option becomes inactive. The next step is shown in Figure 6B.



problem

The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quantity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

Instruction for characterization 1

Quantities: Choose between two different problem conditions:

1. **change quantity** There is a single (only) named quantity in the problem whose value is being changed by a mathematical process(es) that necessarily takes place over a period of time. For example: *Iris has five more dimes today than she had yesterday*. The single/only quantity is Iris's dimes and the number she has is being increased (*more* is the mathematical operator word) by 5 **over time** (yesterday to today)

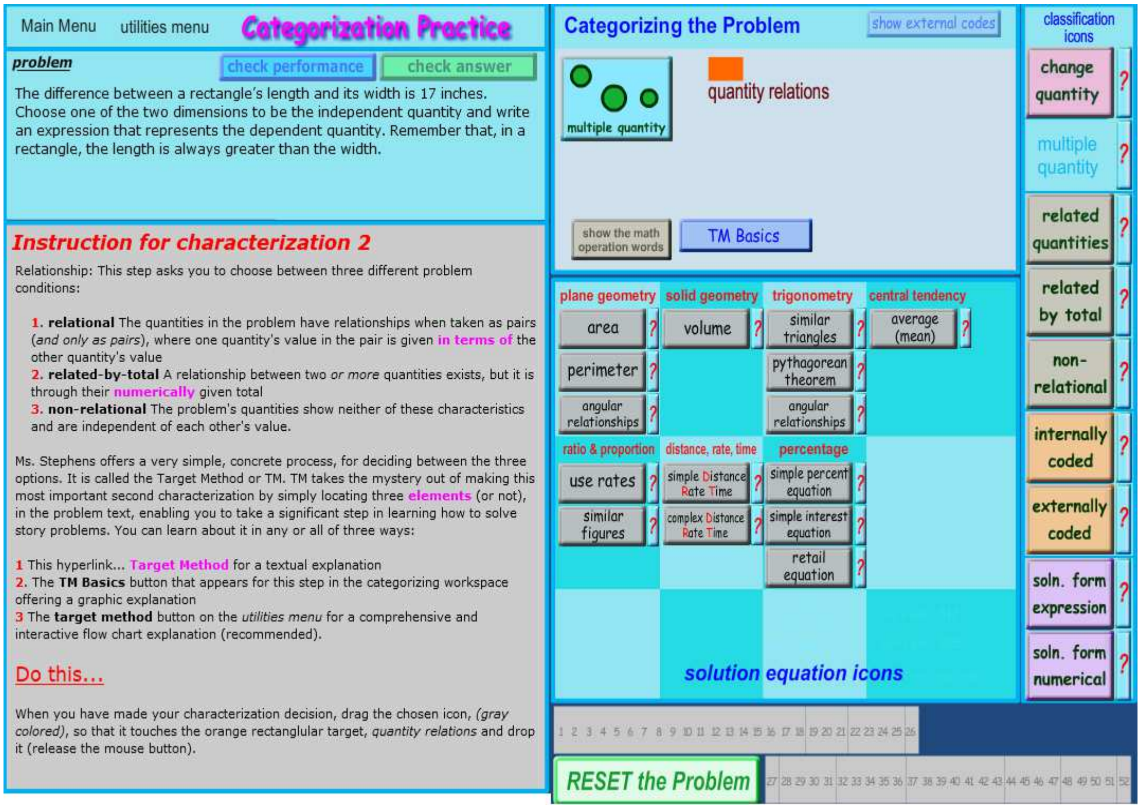
2.**multiple quantity** There are at least two distinctly different quantities in the problem and their values exist **concurrently** (i.e., there is no passage of time between the value established for one quantity and the value established for a second, third or fourth quantity). For example: *Iris has 5 more dimes than Kimberly*. Both Iris and Kimberly have a certain number of dimes **at the same instant in time**

Note: *Additional information regarding any characterization icon on this utility can be had by clicking on the* ***question marked*** *button to the right of the icon. Any of these buttons may be accessed at any time. To dismiss the last instructional graphic just click on the* ***X***

Do this...

When you have made your characterization decision, drag the chosen icon, *(magenta colored)*, so that it touches the orange rectangular target, *quantity number* and drop it (release the mouse button)

Figure 6B. The user clicks on the *multiple quantity* icon as there are three discrete quantities in the problem (length, width, and the difference between them) and begins dragging the icon to the target. The next step is shown in Figure 7.



problem

The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quantity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

Instruction for characterization 2

Relationship: This step asks you to choose between three different problem conditions:

1. **relational** The quantities in the problem have relationships when taken as pairs (*and only as pairs*), where one quantity's value in the pair is given **in terms of** the other quantity's value
2. **related-by-total** A relationship between two *or more* quantities exists, but it is through their **numerically** given total
3. **non-relational** The problem's quantities show neither of these characteristics and are independent of each other's value.

Ms. Stephens offers a very simple, concrete process, for deciding between the three options. It is called the Target Method or TM. TM takes the mystery out of making this most important second characterization by simply locating three **elements** (or not), in the problem text, enabling you to take a significant step in learning how to solve story problems. You can learn about it in any or all of three ways:

1 This hyperlink... **Target Method** for a textual explanation
2. The **TM Basics** button that appears for this step in the categorizing workspace offering a graphic explanation
3 The **target method** button on the *utilities menu* for a comprehensive and interactive flow chart explanation (recommended).

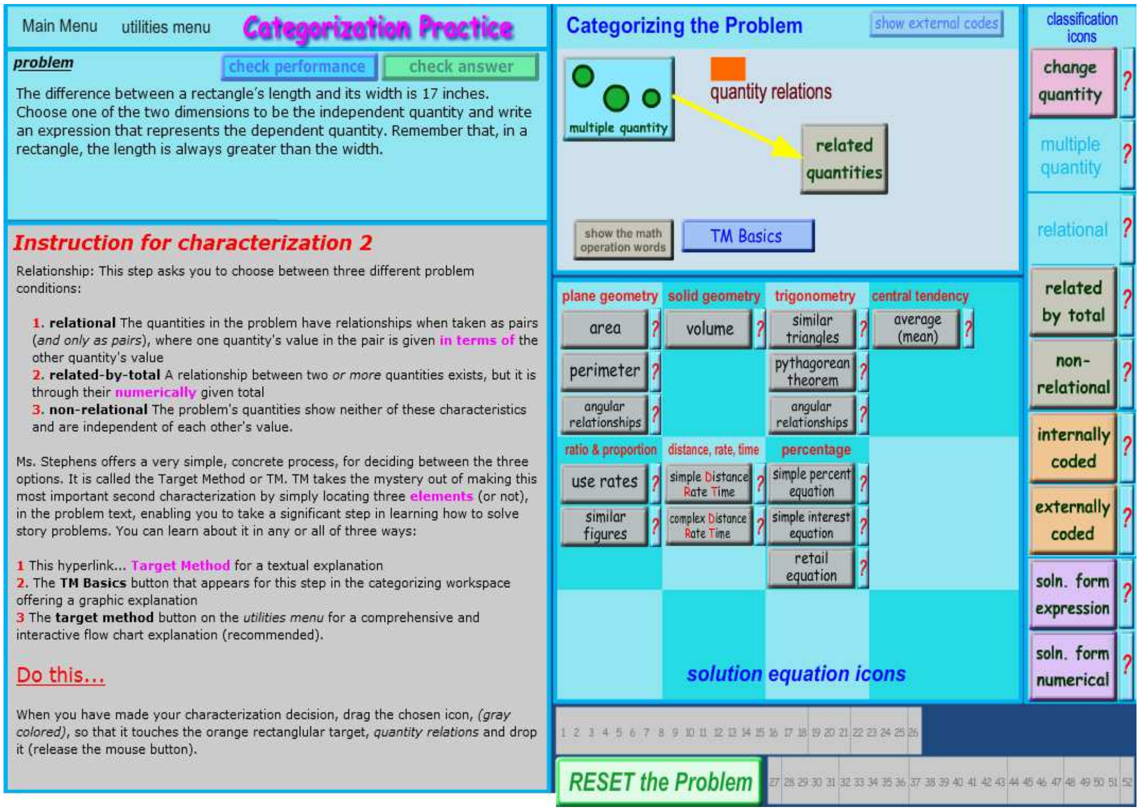
Do this...

When you have made your characterization decision, drag the chosen icon, *(gray colored)*, so that it touches the orange rectanglular target, *quantity relations* and drop it (release the mouse button).

Figure 7. The D-D is completed when the icon touches the target, and the mouse key is released. The second target (*quantity relations*) appears on the Work Space, the relevant instructions show in the Instructions Window*,* and the three icons germane to this characterization are enabled. After reading the instructions, the user may click any or all of these options for additional help:

1. The **show the math operation words** button in the lower-left portion of the workspace. This action highlights in red, the math operation words, if any are present in the problem (see Figure 31, pg. 70). After a certain number of uses, this option becomes inactive.

2. The **TM Basics** button also in the lower-left portion of the workspace. This action presents an animated process for the determination of this characterization (see Figure 23, pg. 62). The use of this option is unlimited.
3. The **Target Method Flowchart** option on the Utilities Menu located in the upper-left portion of the MS-CPU. This action presents a full-screen interactive flowchart that can be used in three different modes to understand how to determine the third characterization (see Figure 24, pg. 63). The use of this option is unlimited. The next step is shown in Figure 8.



The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quantity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

Instruction for characterization 2

Relationship: This step asks you to choose between three different problem conditions:

1. **relational** The quantities in the problem have relationships when taken as pairs (*and only as pairs*), where one quantity's value in the pair is given **in terms of** the other quantity's value
2. **related-by-total** A relationship between two *or more* quantities exists, but it is through their **numerically** given total
3. **non-relational** The problem's quantities show neither of these characteristics and are independent of each other's value.

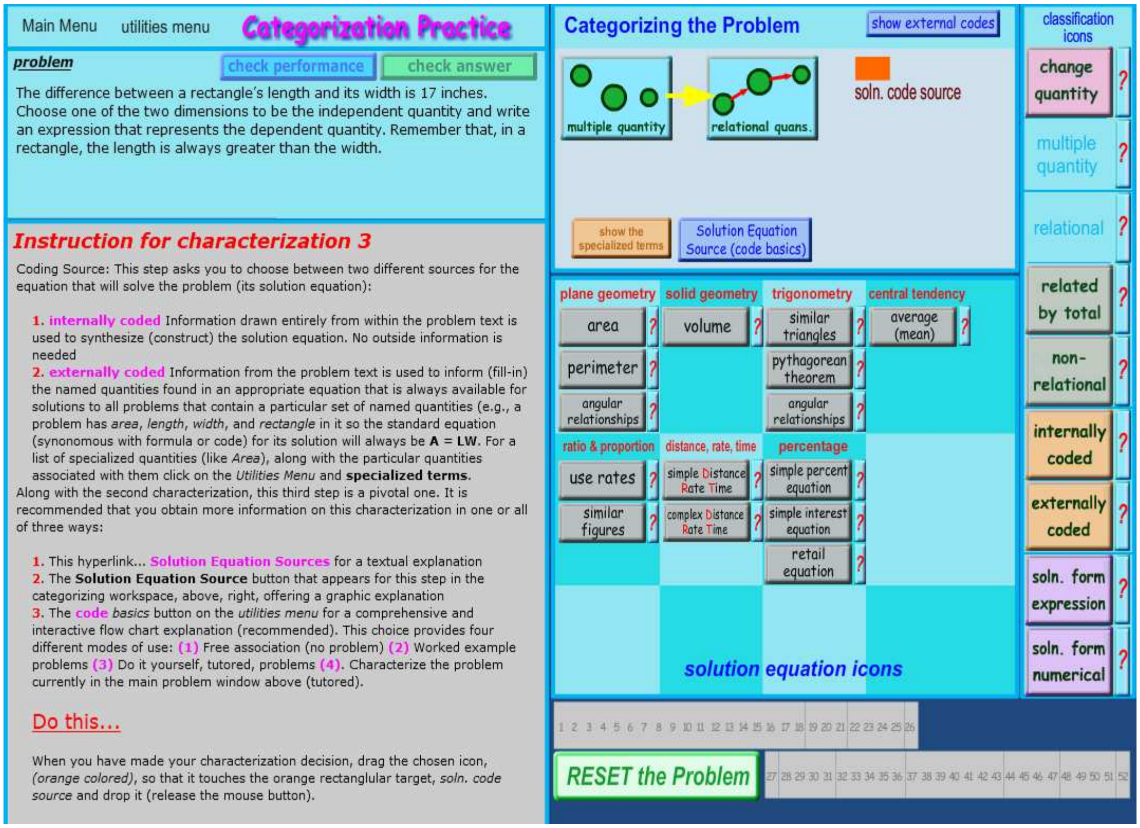
Ms. Stephens offers a very simple, concrete process, for deciding between the three options. It is called the Target Method or TM. TM takes the mystery out of making this most important second characterization by simply locating three **elements** (or not), in the problem text, enabling you to take a significant step in learning how to solve story problems. You can learn about it in any or all of three ways:

1 This hyperlink... **Target Method** for a textual explanation
2. The **TM Basics** button that appears for this step in the categorizing workspace offering a graphic explanation
3 The **target method** button on the *utilities menu* for a comprehensive and interactive flow chart explanation (recommended).

Do this...

When you have made your characterization decision, drag the chosen icon, *(gray colored)*, so that it touches the orange rectanglular target, *quantity relations* and drop it (release the mouse button).

Figure 8. Here, the user clicks on the *related quantities* icon as there is at least one quantity pair where one of the pair's values is dependent on the other (the *difference between length and width*). An arrow visually connects the two characteristics as the user begins dragging the icon to the target and the categorization process advances sequentially. The next step is shown in Figure 9.



problem

The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quantity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

Instruction for characterization 3

Coding Source: This step asks you to choose between two different sources for the equation that will solve the problem (its solution equation):

1. internally coded Information drawn entirely from within the problem text is used to synthesize (construct) the solution equation. No outside information is needed
2. externally coded Information from the problem text is used to inform (fill-in) the named quantities found in an appropriate equation that is always available for solutions to all problems that contain a particular set of named quantities (e.g., a problem has *area*, *length*, *width*, and *rectangle* in it so the standard equation (synonomous with formula or code) for its solution will always be **A = LW**. For a list of specialized quantities (like *Area*), along with the particular quantities associated with them click on the *Utilities Menu* and **specialized terms**.

Along with the second characterization, this third step is a pivotal one. It is recommended that you obtain more information on this characterization in one or all of three ways:

1. This hyperlink... Solution Equation Sources for a textual explanation
2. The **Solution Equation Source** button that appears for this step in the categorizing workspace, above, right, offering a graphic explanation
3. The code *basics* button on the *utilities menu* for a comprehensive and interactive flow chart explanation (recommended). This choice provides four different modes of use: (1) Free association (no problem) (2) Worked example problems (3) Do it yourself, tutored, problems (4). Characterize the problem currently in the main problem window above (tutored).

Do this...

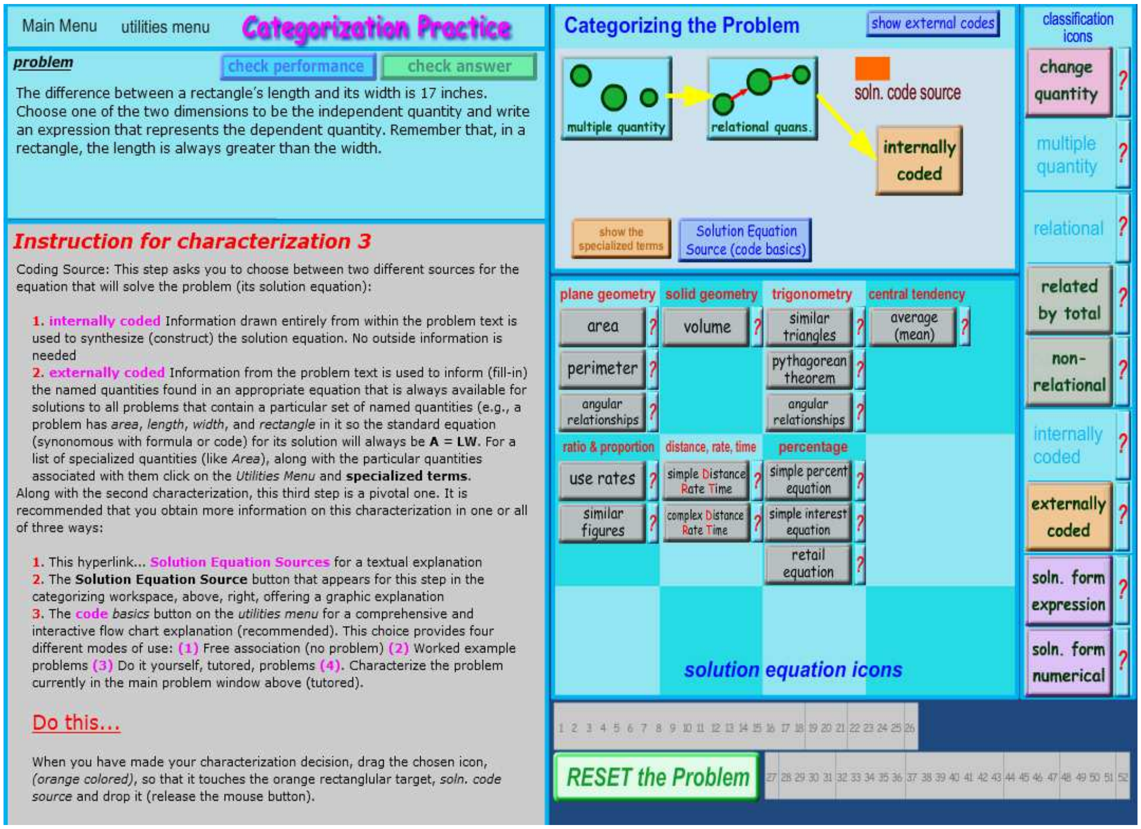
When you have made your characterization decision, drag the chosen icon, *(orange colored)*, so that it touches the orange rectanglular target, *soln. code source* and drop it (release the mouse button).

Figure 9. The D-D is completed when the icon touches the target, and the mouse key is released. Note that the connection between the two characteristics continues to be shown by the yellow arrow. The third target (*solution code source*) appears on the Work Space, the relevant instructions show in the Instructions Window*,* and the two icons germane to this characterization are enabled. After reading the instructions, the user may click any or all of these options for additional help:

1. The **show the specialized terms** button in the lower-left portion of the workspace. This action highlights in red, the special terms (words) that indicate the possible presence of a solution equation (code) that can be sourced externally from the problem. The specialized terms in this problem that would be highlighted are length, width, and rectangle. After a certain number of uses, this option becomes inactive.

2. The **Solution Equation Source (code basics)** button also in the lower-left portion of the workspace. This action presents an animated process for the determination of this characterization. The use of this option is unlimited.
3. The **Solution Source Flowchart** option on the Utilities Menu located in the upper-left portion of the MS-CPU. This action presents a full-screen interactive flowchart that can be used in three different modes to understand how to determine the third characterization. (1) free exploration (2) worked examples (3) do-it-yourself problems. The use of this option is unlimited.

The next step is shown below in Figure 10.



problem

The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quantity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

Instruction for characterization 3

Coding Source: This step asks you to choose between two different sources for the equation that will solve the problem (its solution equation):

1. **internally coded** Information drawn entirely from within the problem text is used to synthesize (construct) the solution equation. No outside information is needed
2. **externally coded** Information from the problem text is used to inform (fill-in) the named quantities found in an appropriate equation that is always available for solutions to all problems that contain a particular set of named quantities (e.g., a problem has *area*, *length*, *width*, and *rectangle* in it so the standard equation (synonomous with formula or code) for its solution will always be **A = LW**. For a list of specialized quantities (like *Area*), along with the particular quantities associated with them click on the *Utilities Menu* and **specialized terms**.

Along with the second characterization, this third step is a pivotal one. It is recommended that you obtain more information on this characterization in one or all of three ways:

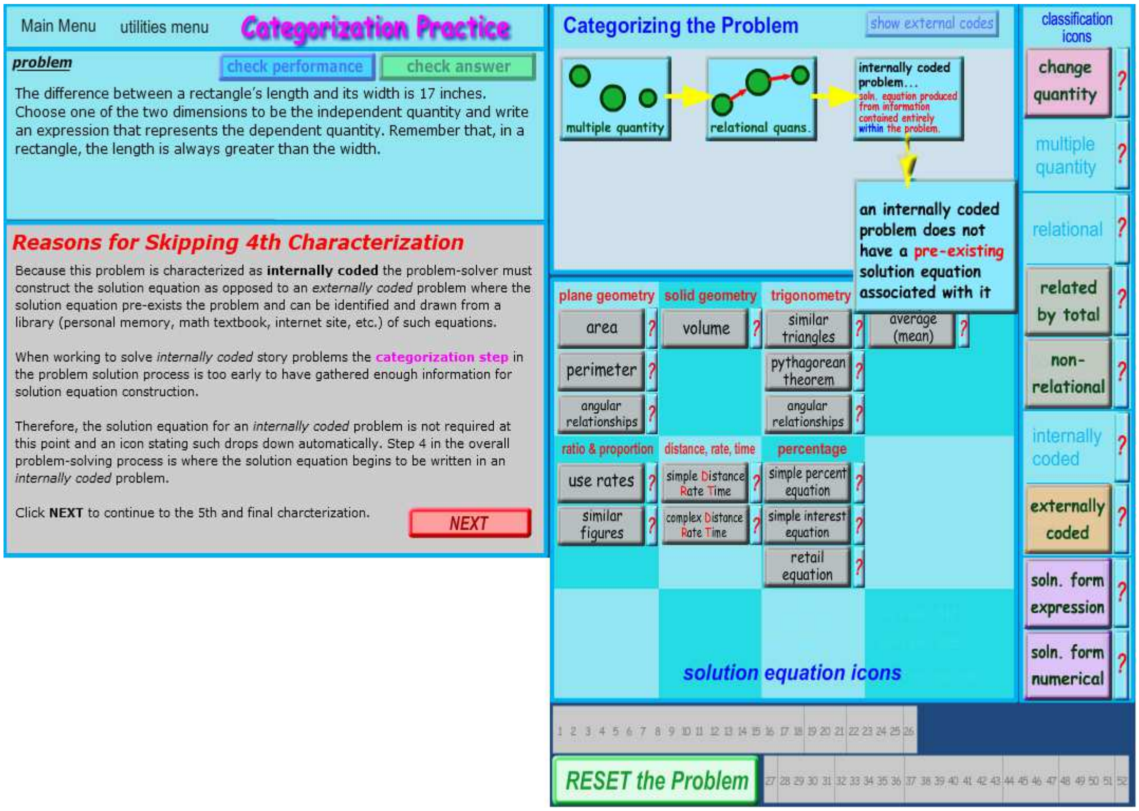
1. This hyperlink... **Solution Equation Sources** for a textual explanation
2. The **Solution Equation Source** button that appears for this step in the categorizing workspace, above, right, offering a graphic explanation
3. The **code** *basics* button on the *utilities menu* for a comprehensive and interactive flow chart explanation (recommended). This choice provides four different modes of use: (1) Free association (no problem) (2) Worked example problems (3) Do it yourself, tutored, problems (4). Characterize the problem currently in the main problem window above (tutored).

Do this...

When you have made your characterization decision, drag the chosen icon, *(orange colored)*, so that it touches the orange rectanglular target, *soln. code source* and drop it (release the mouse button).

Figure 10. There are three specialized terms, length, width, and rectangle. However, either area or perimeter is required to complete the externally sourced solution equation for a rectangle of A = lw or P = 2l + 2w. Since neither is present, the third problem characteristic is *internally coded*. An arrow visually connects the two characteristics as the user begins dragging the icon to

the target and the categorization process advances sequentially. The next step is shown in Figure 11.



The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quantity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

Reasons for Skipping 4th Characterization

Because this problem is characterized as **internally coded** the problem-solver must construct the solution equation as opposed to an *externally coded* problem where the solution equation pre-exists the problem and can be identified and drawn from a library (personal memory, math textbook, internet site, etc.) of such equations.

When working to solve *internally coded* story problems the **categorization step** in the problem solution process is too early to have gathered enough information for solution equation construction.

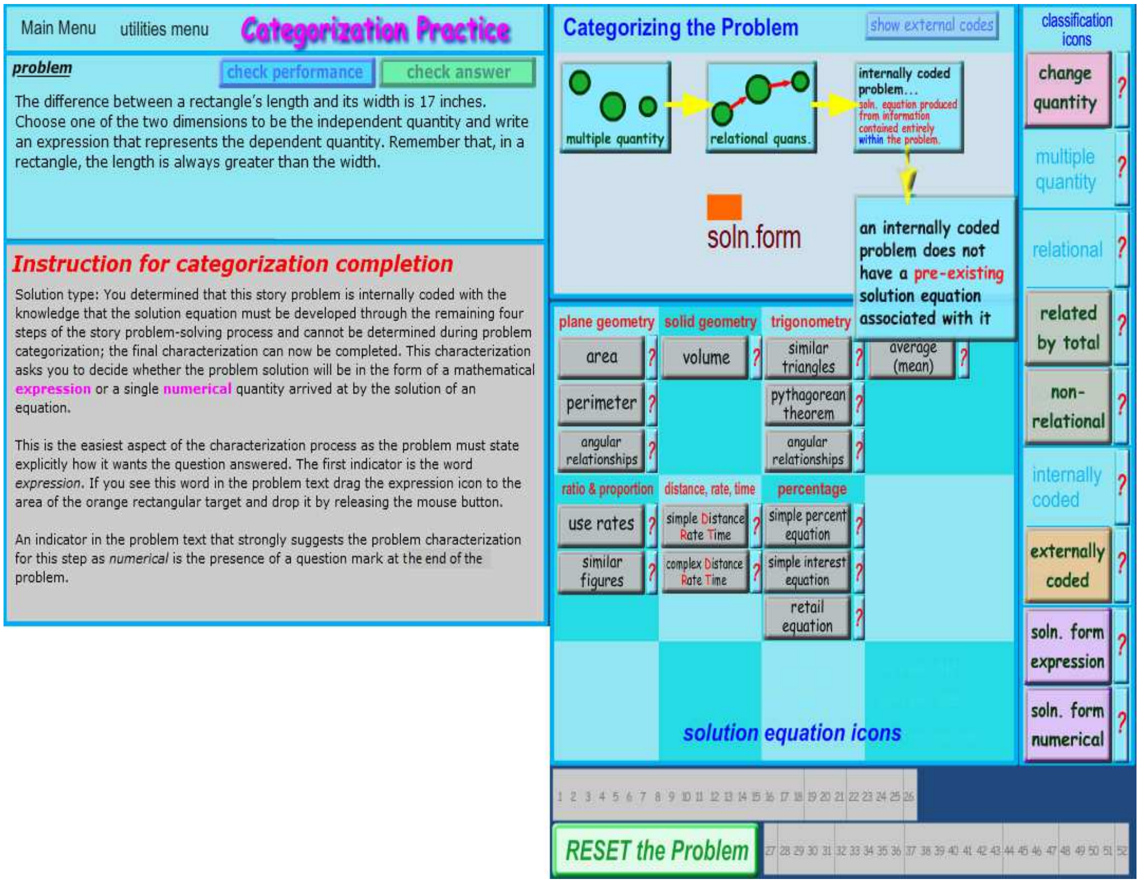
Therefore, the solution equation for an *internally coded* problem is not required at this point and an icon stating such drops down automatically. Step 4 in the overall problem-solving process is where the solution equation begins to be written in an *internally coded* problem.

Click **NEXT** to continue to the 5th and final charcterization.

NEXT
Categorizing the Problem
show external codes
multiple quantity
relational quans.
internally coded problem...
soln. equation produced from information contained entirely within the problem.

an internally coded problem does not have a pre-existing solution equation associated with it

Figure 11. The D-D is completed when the icon touches the target, and the mouse key is released. Note that the connection between the two characteristics continues to be shown by the yellow arrow. The fourth characterization (*solution code*) icon automatically drops down in the Work Space because the third problem characteristic is *internally coded*, meaning there are no pre-existing solution equations with length and width as the only two quantities in such an equation. The reasoning behind the automatic drop is in the Instructions Window*,* and all of the solution equation icons in the bottom half of the UI are disabled. The next step is shown in Figure 12 after the red ***NEXT*** button is clicked.



The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quantity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

soln. equation produced from information contained entirely within the problem.

an internally coded problem does not have a pre-existing solution equation associated with it

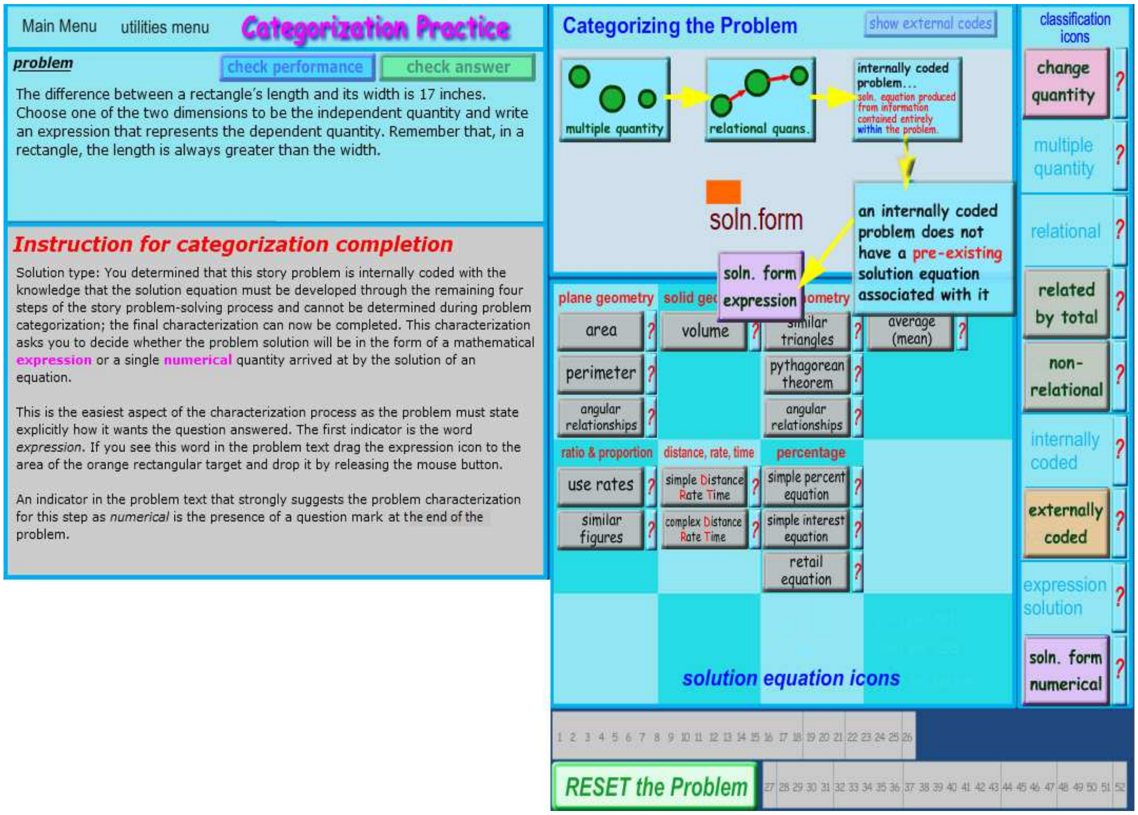
Instruction for categorization completion

Solution type: You determined that this story problem is internally coded with the knowledge that the solution equation must be developed through the remaining four steps of the story problem-solving process and cannot be determined during problem categorization; the final characterization can now be completed. This characterization asks you to decide whether the problem solution will be in the form of a mathematical **expression** or a single **numerical** quantity arrived at by the solution of an equation.

This is the easiest aspect of the characterization process as the problem must state explicitly how it wants the question answered. The first indicator is the word *expression*. If you see this word in the problem text drag the expression icon to the area of the orange rectangular target and drop it by releasing the mouse button.

An indicator in the problem text that strongly suggests the problem characterization for this step as *numerical* is the presence of a question mark at the end of the problem.

Figure 12. The fifth target (*solution form*) appears on the Work Space, the relevant instructions show in the Instructions Window*,* and the two characteristic icons germane to this categorization step are enabled. Before choosing a *solution form* icon, the user may click the question mark to the right of either icon for a detailed explanation of the characteristic. The next step is shown below in Figure 13.



The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quantity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

internally coded problem... soln. equation produced from information contained entirely within the problem.

an internally coded problem does not have a pre-existing solution equation associated with it

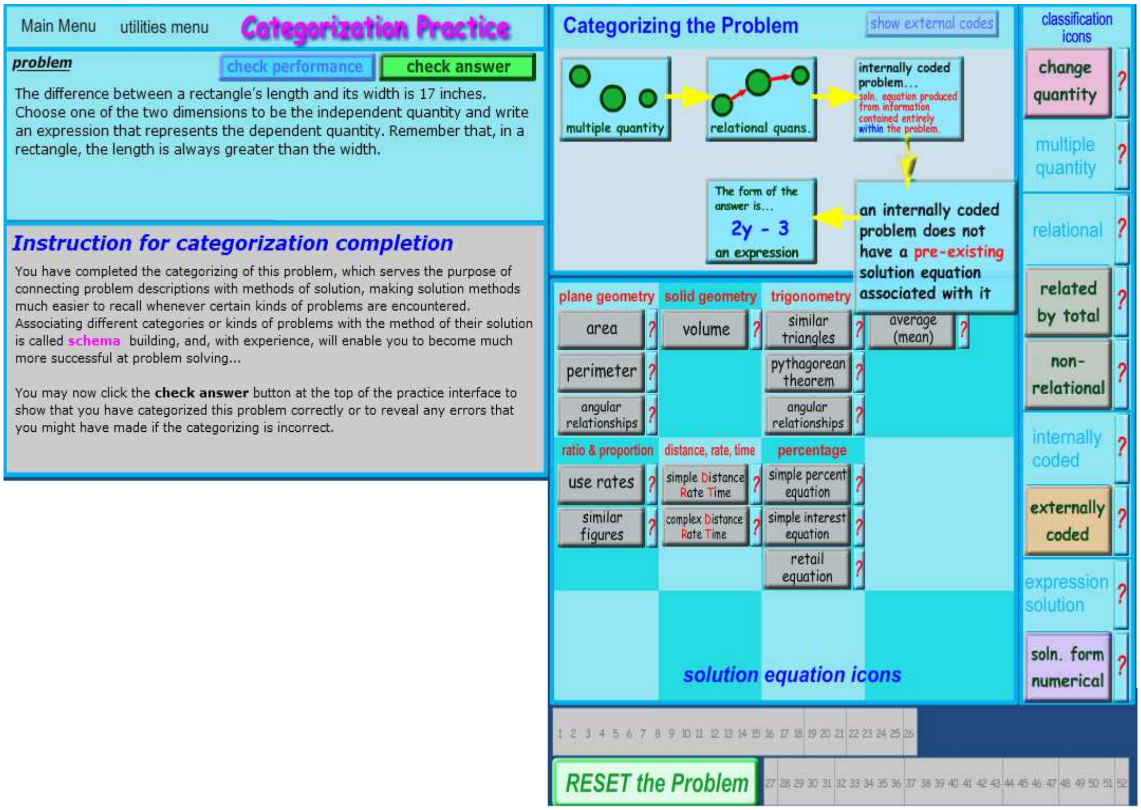
Instruction for categorization completion

Solution type: You determined that this story problem is internally coded with the knowledge that the solution equation must be developed through the remaining four steps of the story problem-solving process and cannot be determined during problem categorization; the final characterization can now be completed. This characterization asks you to decide whether the problem solution will be in the form of a mathematical **expression** or a single **numerical** quantity arrived at by the solution of an equation.

This is the easiest aspect of the characterization process as the problem must state explicitly how it wants the question answered. The first indicator is the word *expression*. If you see this word in the problem text drag the expression icon to the area of the orange rectangular target and drop it by releasing the mouse button.

An indicator in the problem text that strongly suggests the problem characterization for this step as *numerical* is the presence of a question mark at the end of the problem.

Figure 13. After reading the instructions, the user clicks on the appropriate icon and begins dragging the *soln. form expression* icon (the problem asks for the answer in the form of an expression) to the Work Space target. Note that there is an arrow visually connecting the two characteristics as the categorization process advances sequentially. The next step is shown below in Figure 14.



problem

The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quantity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

Instruction for categorization completion

You have completed the categorizing of this problem, which serves the purpose of connecting problem descriptions with methods of solution, making solution methods much easier to recall whenever certain kinds of problems are encountered. Associating different categories or kinds of problems with the method of their solution is called schema building, and, with experience, will enable you to become much more successful at problem solving...

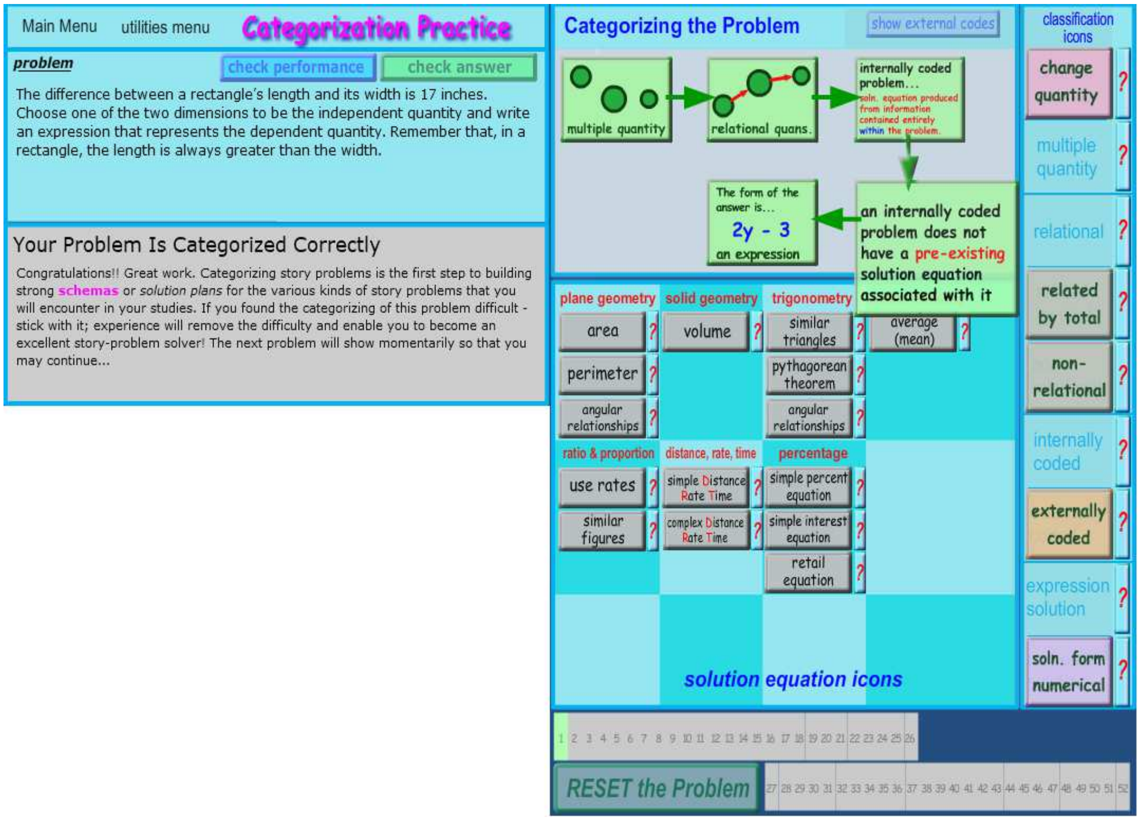
You may now click the **check answer** button at the top of the practice interface to show that you have categorized this problem correctly or to reveal any errors that you might have made if the categorizing is incorrect.

internally coded problem... soln. equation produced from information contained entirely within the problem.

The form of the answer is... 2y - 3 an expression

an internally coded problem does not have a pre-existing solution equation associated with it

Figure 14. The D_D is completed when the icon touches the target, and the mouse key is released. The connection between the two characteristics continues to be shown by the yellow arrow, and the check answer button is now enabled. Note that the flowchart references are no longer relevant as the categorization of the problem is complete. Figure 15 shows the MS-CPU after the check answer has been clicked.



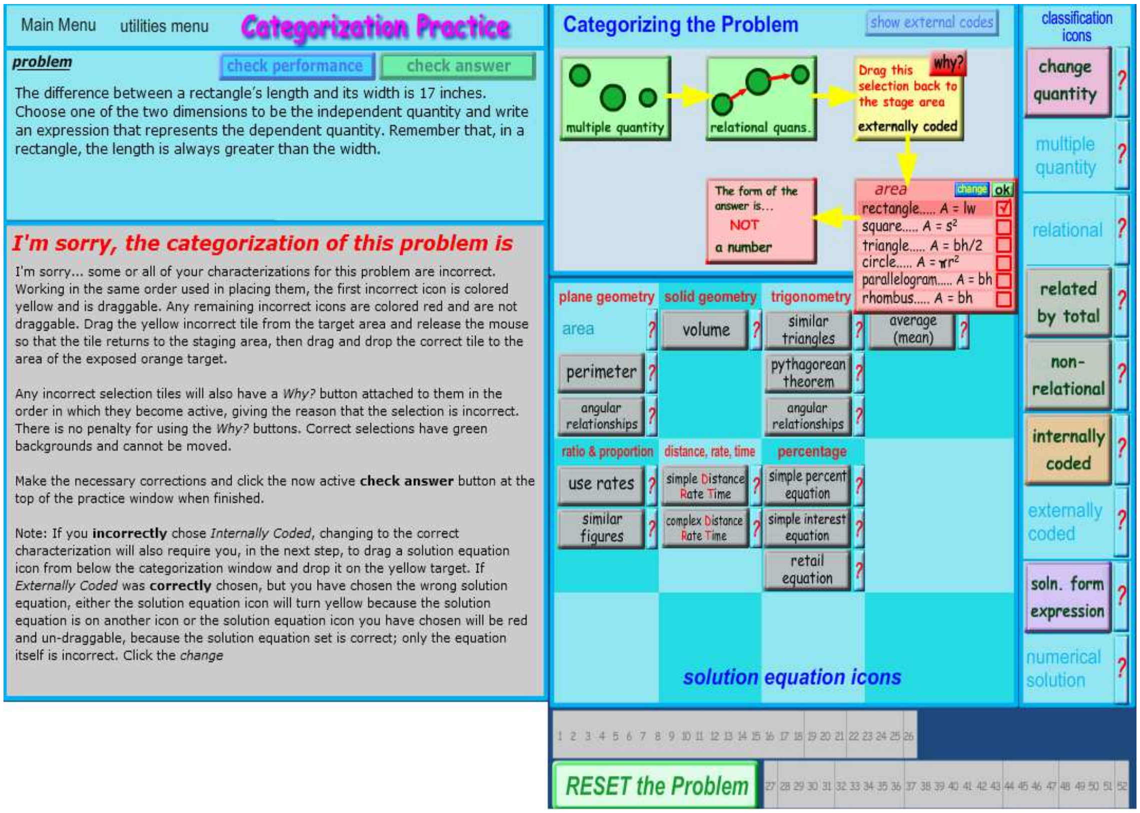
The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quantity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

Your Problem Is Categorized Correctly

Congratulations!! Great work. Categorizing story problems is the first step to building strong schemas or *solution plans* for the various kinds of story problems that you will encounter in your studies. If you found the categorizing of this problem difficult - stick with it; experience will remove the difficulty and enable you to become an excellent story-problem solver! The next problem will show momentarily so that you may continue...

Figure 15. If the selected characteristics of the problem are correct, the progress bar is updated by coloring the first block in the progress bar green, individual characteristics information is updated, all icons turn to green, and a congratulatory message is shown in the Instructions Window*.* There is a delay, giving time enough for the user to read the message before the instructions for the next problem appear.

If, on the other hand, some or all of the characteristics are incorrect the MSASPT colors each incorrect icon red except for the first (in the order of characterizations) which is yellow and has both a *why* button and a message to drag the icon back to its stage as shown in Figure 16.



problem

The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quantity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

I'm sorry, the categorization of this problem is

I'm sorry... some or all of your characterizations for this problem are incorrect. Working in the same order used in placing them, the first incorrect icon is colored yellow and is draggable. Any remaining incorrect icons are colored red and are not draggable. Drag the yellow incorrect tile from the target area and release the mouse so that the tile returns to the staging area, then drag and drop the correct tile to the area of the exposed orange target.

Any incorrect selection tiles will also have a *Why?* button attached to them in the order in which they become active, giving the reason that the selection is incorrect. There is no penalty for using the *Why?* buttons. Correct selections have green backgrounds and cannot be moved.

Make the necessary corrections and click the now active **check answer** button at the top of the practice window when finished.

Note: If you **incorrectly** chose *Internally Coded*, changing to the correct characterization will also require you, in the next step, to drag a solution equation icon from below the categorization window and drop it on the yellow target. If *Externally Coded* was **correctly** chosen, but you have chosen the wrong solution equation, either the solution equation icon will turn yellow because the solution equation is on another icon or the solution equation icon you have chosen will be red and un-draggable, because the solution equation set is correct; only the equation itself is incorrect. Click the *change*

Drag this selection back to the stage area

The form of the answer is... NOT a number

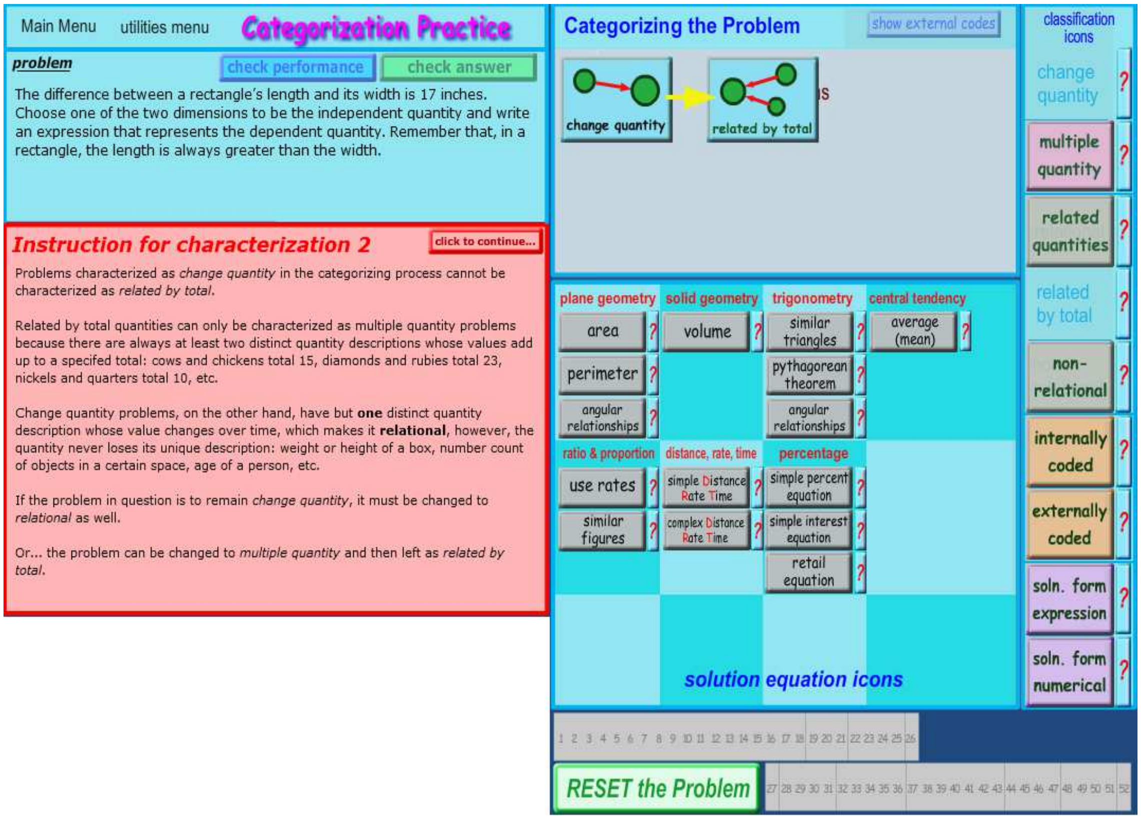
Figure 16. Since the characteristics are dichotomous/trichotomous there is no reason to hint the correct answer; the reason for the error is supplied on demand by clicking a **why** button. Once the first error is remediated, if there are any further errors, the next icon changes color to yellow, and the process is repeated until all incorrect characteristics have been corrected. The check answer button is again clicked, and the categorization is evaluated. Each incorrect response is appropriately recorded in the student model and presented on demand at certain intervals in the problem categorization session.

When a story problem is characterized as *change quantity,* certain constraints apply that serve to make the categorization of these problems unitary except for the *solution form/type* characterization which, of course, is always dichotomous. A *change quantity* problem must:

- Be characterized as *relational* because it would be impossible for the new quantity in such a problem not to be dependent for its value on the original quantity's value, which is the very definition of *relational*.

- Be characterized as *internally coded* because there are no pre-existing solution equations for change quantity problems.
- Be characterized as having no solution set yet because it is *internally coded.*

The dichotomous nature of the categorization is a result of the two different possibilities for *solution form/type*. Once the *change quantity* characterization is made, any deviation results in an error message. For example, if the user characterized a *change quantity* problem as *related-by-total* the UI would appear as in Figure 17:



The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quantity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

Instruction for characterization 2

Problems characterized as *change quantity* in the categorizing process cannot be characterized as *related by total.*

Related by total quantities can only be characterized as multiple quantity problems because there are always at least two distinct quantity descriptions whose values add up to a specifed total: cows and chickens total 15, diamonds and rubies total 23, nickels and quarters total 10, etc.

Change quantity problems, on the other hand, have but **one** distinct quantity description whose value changes over time, which makes it **relational**, however, the quantity never loses its unique description: weight or height of a box, number count of objects in a certain space, age of a person, etc.

If the problem in question is to remain *change quantity*, it must be changed to *relational* as well.

Or... the problem can be changed to *multiple quantity* and then left as *related by total.*

Figure 17. The instructions window contains information about the mutual exclusiveness of *change quantity* and *relate-by-total* characterizations in the same problem. Once the **click to continue** button is clicked the icons automatically return to the stage. Similar messages appear when *change quantity* is used in conjunction with a *non-relational* characteristic and when it is used with *externally coded* in the third characterization. It is important to note that the decisions to immediately identify errors in certain parts of the MSASPT Categorization process, rather than

wait until the process completes (the prescriptive sequence) derives from the dichotomous nature of the characterizations and the unexpected interjection that I suggest, may well be included among the best practices for teaching a process.

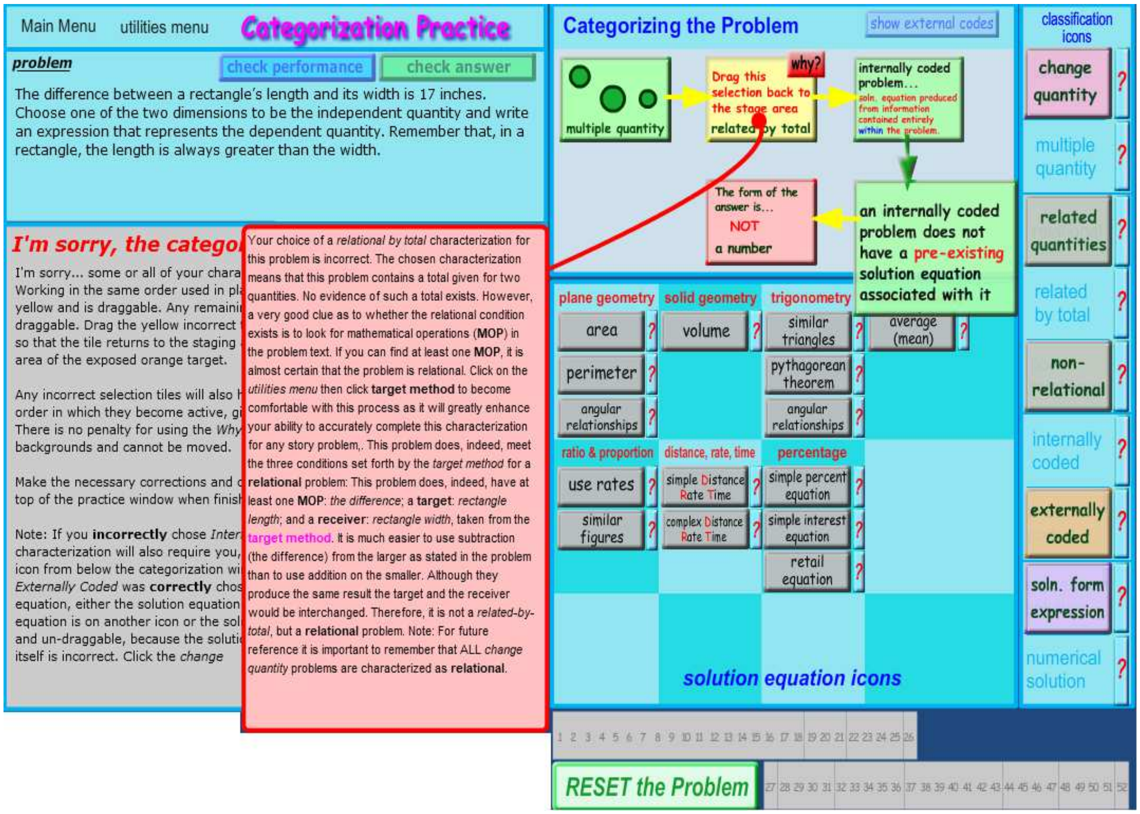
The MSASPT Categorization process institutes a unique method for identifying a story problem through the enumeration of the problem's characteristics. This process has the dual purpose of not only using the characteristics as schema constituents and retrieval cues but also using them to prompt actions that will help the problem-solver as s/he formulates a solution plan culminating in a solution equation. This paper will now turn its attention to the MSASPT Categorization Step Practice Program (CS-PP) features that facilitate the pedagogy and enhance its effectiveness.

The MSASPT Categorization Step Features:

One of the most important features of an ITS is its ability to provide timely and accurate hints for the problem-solver when they make a mistake or have clearly left any possible solution track. It is largely accepted that hinting is done at three different levels:

1) A pointing hint that may serve to jog the problem solver's memory.
2) A teaching hint that re-teaches, perhaps from a different perspective, a principle or concept that should be known.
3) A bottom out hint that furnishes the answer.

Because of the dichotomous/trichotomous nature of the characteristics that make up the categorization process, three hint levels are unnecessary. Instead, the student is offered a simple why button which, when clicked, explains the characteristic and provides the reasoning that supports the other choice as Figure 18 illustrates.



problem

The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quantity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

I'm sorry, the catego

I'm sorry... some or all of your chara
Working in the same order used in pl
yellow and is draggable. Any remaini
draggable. Drag the yellow incorrect
so that the tile returns to the staging
area of the exposed orange target.

Any incorrect selection tiles will also
order in which they become active, g
There is no penalty for using the *Why*
backgrounds and cannot be moved.

Make the necessary corrections and c
top of the practice window when finis

Note: If you **incorrectly** chose *Inter*
characterization will also require you,
icon from below the categorization wi
Externally Coded was **correctly** chos
equation, either the solution equation
equation is on another icon or the sol
and un-draggable, because the soluti
itself is incorrect. Click the *change*

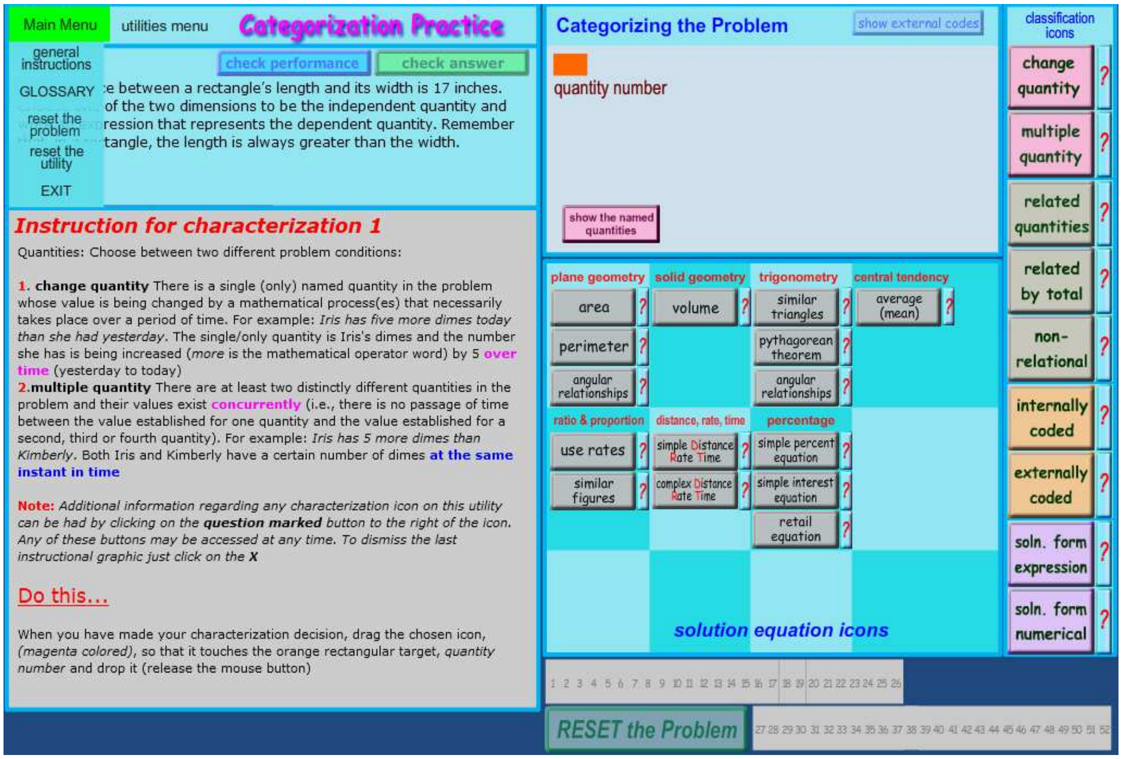
Your choice of a *relational by total* characterization for this problem is incorrect. The chosen characterization means that this problem contains a total given for two quantities. No evidence of such a total exists. However, a very good clue as to whether the relational condition exists is to look for mathematical operations (**MOP**) in the problem text. If you can find at least one **MOP**, it is almost certain that the problem is relational. Click on the *utilities menu* then click **target method** to become comfortable with this process as it will greatly enhance your ability to accurately complete this characterization for any story problem,. This problem does, indeed, meet the three conditions set forth by the *target method* for a **relational** problem: This problem does, indeed, have at least one **MOP**: *the difference*; a **target**: *rectangle length*; and a **receiver**: *rectangle width*, taken from the target method. It is much easier to use subtraction (the difference) from the larger as stated in the problem than to use addition on the smaller. Although they produce the same result the target and the receiver would be interchanged. Therefore, it is not a *related-by-total*, but a **relational** problem. Note: For future reference it is important to remember that ALL *change quantity* problems are characterized as **relational**.

Figure 18. The **why** button has been clicked for the second characterization error (the fifth characterization is also in error) and a message addressing the mistake appears. When the icon is dragged back to the stage, the message disappears, and the pertinent icons are enabled for dragging. A list of additional features along with an explanation of their operation in service of learning how to categorize story problems follows:

1) Removable step instructions – as the student moves through the problem categorization experience several measures of competency are being submitted to the categorization student model. These measures are point-based and are used by the tutor to determine whether there is a need for textual instruction. If it is determined that no need exists, the program stops giving the instructions. However, the student can make that decision earlier as well as decide to turn them back on. Since the measure of competency or acquisition of knowledge components is point-based, written into the evaluation system are bonus points for not having the instructions turned on. This provides the learner with a good measure of

control over the speed of their problem-solving progress from the acquisition of more points (without instructions) or fewer points (with instructions).

2) Main Menu – located in the upper-left corner of the MS-CPU, a drop-down menu is provided to access the various operational features of the system as shown in Figure 19.



:e between a rectangle's length and its width is 17 inches. of the two dimensions to be the independent quantity and ression that represents the dependent quantity. Remember tangle, the length is always greater than the width.

Instruction for characterization 1

Quantities: Choose between two different problem conditions:

1. **change quantity** There is a single (only) named quantity in the problem whose value is being changed by a mathematical process(es) that necessarily takes place over a period of time. For example: *Iris has five more dimes today than she had yesterday*. The single/only quantity is Iris's dimes and the number she has is being increased (*more* is the mathematical operator word) by 5 **over time** (yesterday to today)
2.**multiple quantity** There are at least two distinctly different quantities in the problem and their values exist **concurrently** (i.e., there is no passage of time between the value established for one quantity and the value established for a second, third or fourth quantity). For example: *Iris has 5 more dimes than Kimberly*. Both Iris and Kimberly have a certain number of dimes **at the same instant in time**

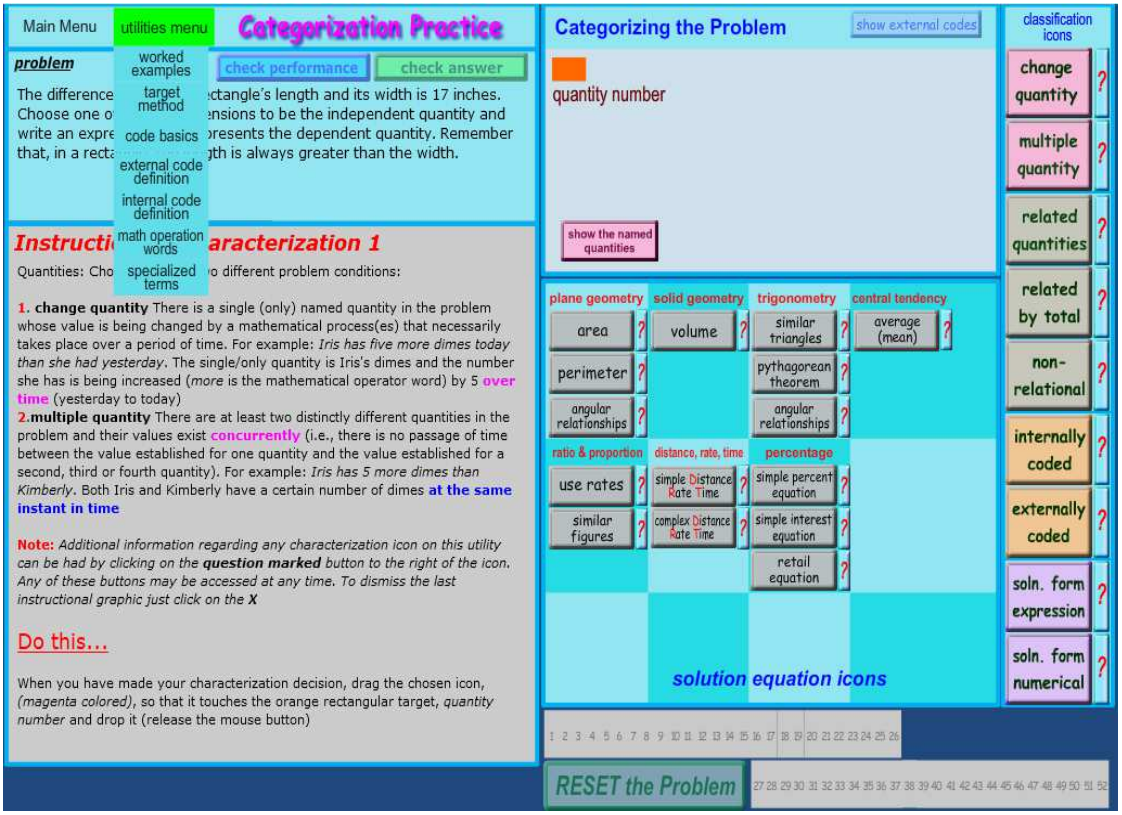
Note: *Additional information regarding any characterization icon on this utility can be had by clicking on the* ***question marked*** *button to the right of the icon. Any of these buttons may be accessed at any time. To dismiss the last instructional graphic just click on the* ***X***

Do this...

When you have made your characterization decision, drag the chosen icon, *(magenta colored)*, so that it touches the orange rectangular target, *quantity number* and drop it (release the mouse button)

Figure 19. The operational features of the system on the Main Menu are the General Instructions, a Glossary of important terms and phrases, an option to reset the problem, an option to reset the entire utility, and an Exit selection.

3) Utilities Menu – located to the right of the Main Menu of the MS-CPU, a drop-down menu is provided the access the major utilities available for additional explanation and practice as shown in Figure 20.



problem

The difference ctangle's length and its width is 17 inches. Choose one o ensions to be the independent quantity and write an expre presents the dependent quantity. Remember that, in a recta gth is always greater than the width.

Instructi aracterization 1

Quantities: Cho o different problem conditions:

1. **change quantity** There is a single (only) named quantity in the problem whose value is being changed by a mathematical process(es) that necessarily takes place over a period of time. For example: *Iris has five more dimes today than she had yesterday*. The single/only quantity is Iris's dimes and the number she has is being increased (*more* is the mathematical operator word) by 5 **over time** (yesterday to today)
2.**multiple quantity** There are at least two distinctly different quantities in the problem and their values exist **concurrently** (i.e., there is no passage of time between the value established for one quantity and the value established for a second, third or fourth quantity). For example: *Iris has 5 more dimes than Kimberly*. Both Iris and Kimberly have a certain number of dimes **at the same instant in time**

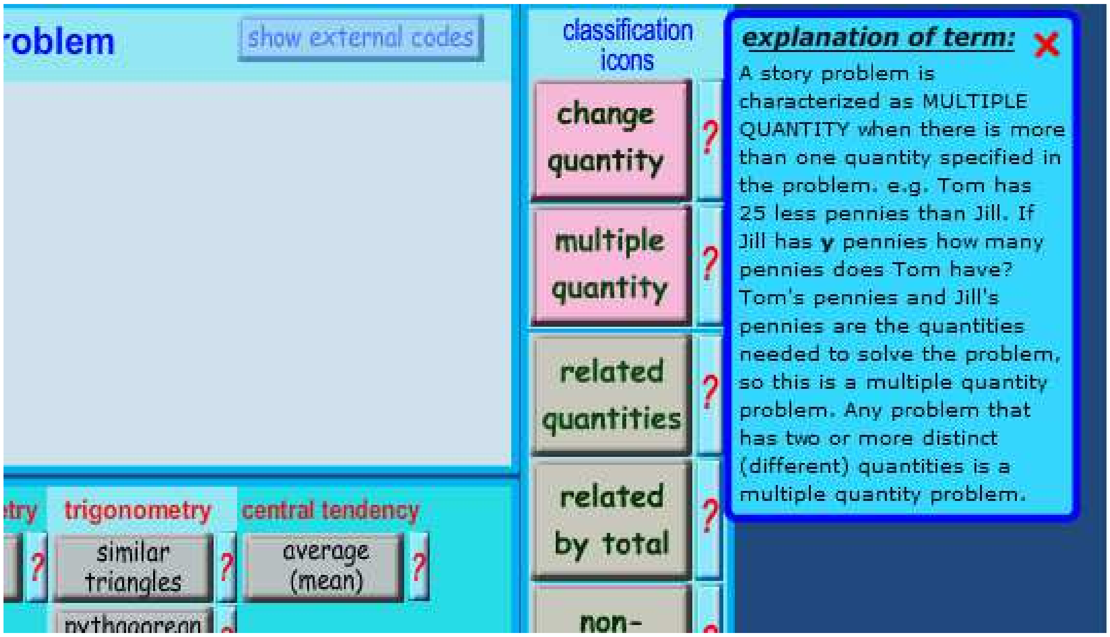
Note: *Additional information regarding any characterization icon on this utility can be had by clicking on the* ***question marked*** *button to the right of the icon. Any of these buttons may be accessed at any time. To dismiss the last instructional graphic just click on the* ***X***

Do this...

When you have made your characterization decision, drag the chosen icon, *(magenta colored)*, so that it touches the orange rectangular target, *quantity number* and drop it (release the mouse button)

Figure 20. The major utilities available on the Utilities Menu are Worked Examples prominently place in the first position, Target Method utility for help with the second characterization, Solution Equation Sources for help with the third characterization, just-in-time internal code definition, just-in-time external code definition, mathematical operation words, and a list of specialized terms and their associated terms.

4) Clickables for explanations of characteristics – to the right of every characteristic icon is a rectangular area labeled with a red question mark (?). Shown below in Figure 21.

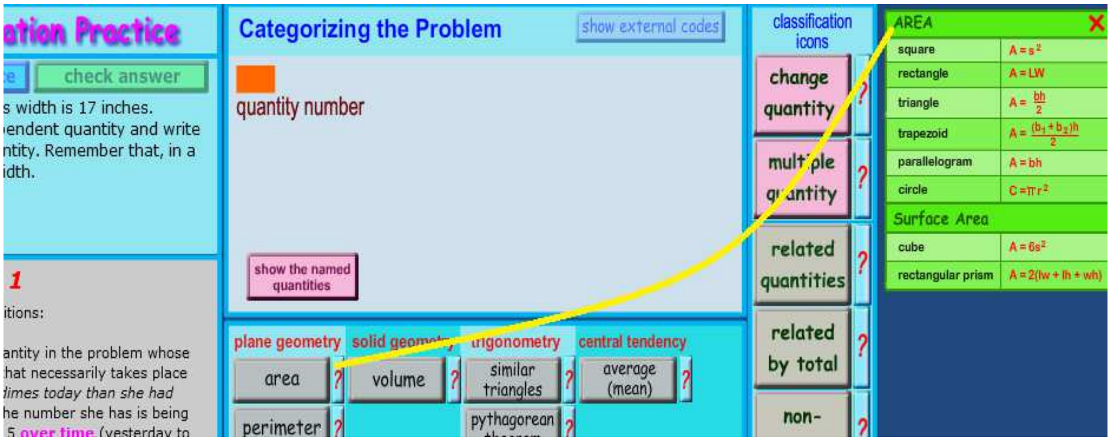


explanation of term:

A story problem is characterized as MULTIPLE QUANTITY when there is more than one quantity specified in the problem. e.g. Tom has 25 less pennies than Jill. If Jill has **y** pennies how many pennies does Tom have? Tom's pennies and Jill's pennies are the quantities needed to solve the problem, so this is a multiple quantity problem. Any problem that has two or more distinct (different) quantities is a multiple quantity problem.

Figure 21. When the button-shaped area marked by a question mark is clicked, a detailed explanation is presented informing the learner of the qualifications that a story problem must possess to be characterized in such a manner (multiple quantity).

5) Clickable areas for explanations of external solution equation families or sets - to the right of every external solution equation family icon is a rectangular area labeled with a red question mark (?). Shown below in Figure 22.



s width is 17 inches.
endent quantity and write
ntity. Remember that, in a
idth.

1

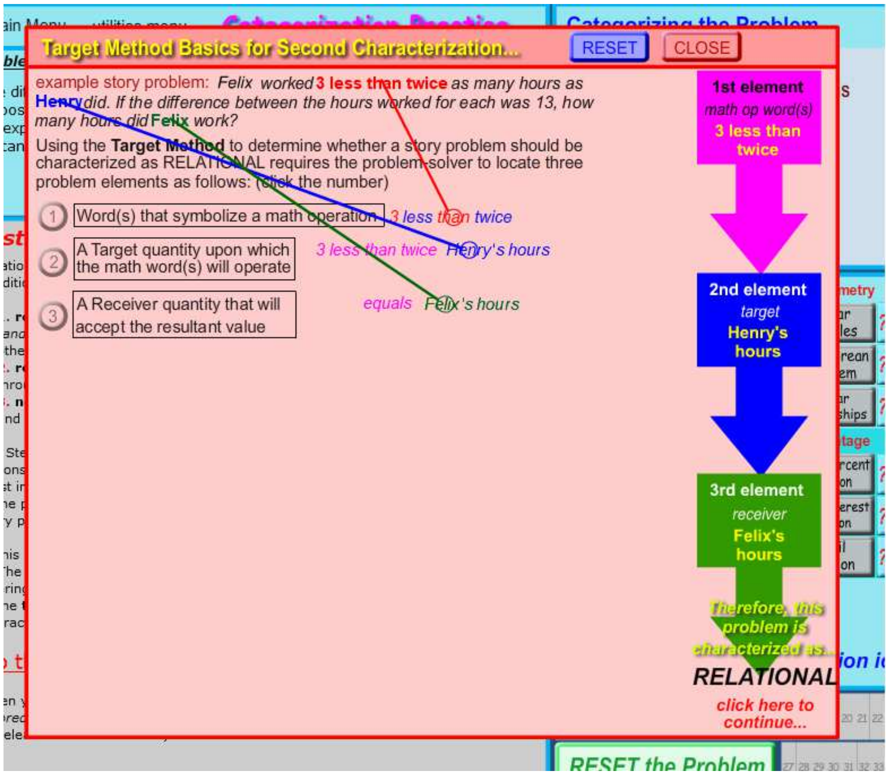
itions:

antity in the problem whose
hat necessarily takes place
times today than she had
he number she has is being
5 over time (yesterday to

AREA	
square	$A = s^2$
rectangle	$A = LW$
triangle	$A = \frac{bh}{2}$
trapezoid	$A = \frac{(b_1 + b_2)h}{2}$
parallelogram	$A = bh$
circle	$C = \pi r^2$
Surface Area	
cube	$A = 6s^2$
rectangular prism	$A = 2(lw + lh + wh)$

Figure 22. When clicked, a detailed explanation is presented that displays the solution equations that exist in a certain family of equations (e.g., plane geometry area: rectangle, square, parallelogram, rhombus, etc.), with an illustration if appropriate.

6) Target Method Basics for determining second characterization graphic – Clicking on the TM Basics button presents an animated graphic for instruction on how to use the Target Method shown in Figure 23.



Target Method Basics for Second Characterization...

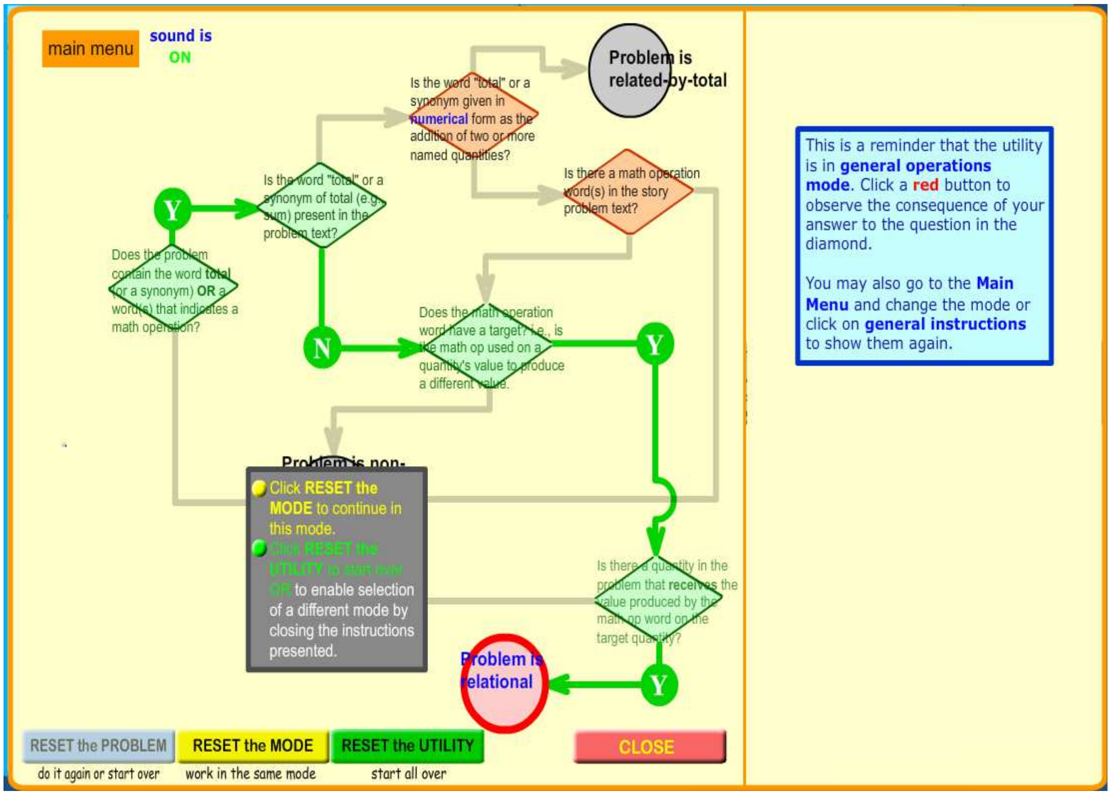
example story problem: *Felix worked* **3 less than twice** *as many hours as* **Henry** *did. If the difference between the hours worked for each was 13, how many hours did* **Felix** *work?*

Using the **Target Method** to determine whether a story problem should be characterized as RELATIONAL requires the problem-solver to locate three problem elements as follows: (click the number)

1. Word(s) that symbolize a math operation
2. A Target quantity upon which the math word(s) will operate
3. A Receiver quantity that will accept the resultant value

Figure 23. This graphic animation demonstrates a concrete process where the student uses clues within the story problem to help determine whether it should be characterized as *relational*, *related-by-total*, or *non-relational*. The animation state shown depicts the determination of a *relational* characterization. Further use shows *related-by-total* and *non-relational* determinations as well as how to use concrete articulation in the process. Full use of the Target Method enables the problem-solver to determine the relational structure of the problem as explained in Section IV.

7) An interactive flowchart demonstrates the Target Method (TM) is shown in Figure 24.


main menu
sound is ON
Problem is related-by-total
Is the word "total" or a synonym given in numerical form as the addition of two or more named quantities?
Is there a math operation word(s) in the story problem text?
Y
Is the word "total" or a synonym of total (e.g. sum) present in the problem text?
Does the problem contain the word total (or a synonym) OR a word(s) that indicates a math operation?
N
Does the math operation word have a target? i.e., is the math op used on a quantity's value to produce a different value.
Y
Problem is non-
Is there a quantity in the problem that receives the value produced by the math op word on the target quantity?
Problem is relational
Y
RESET the PROBLEM
do it again or start over
RESET the MODE
work in the same mode
RESET the UTILITY
start all over
CLOSE

Click **RESET the MODE** to continue in this mode.

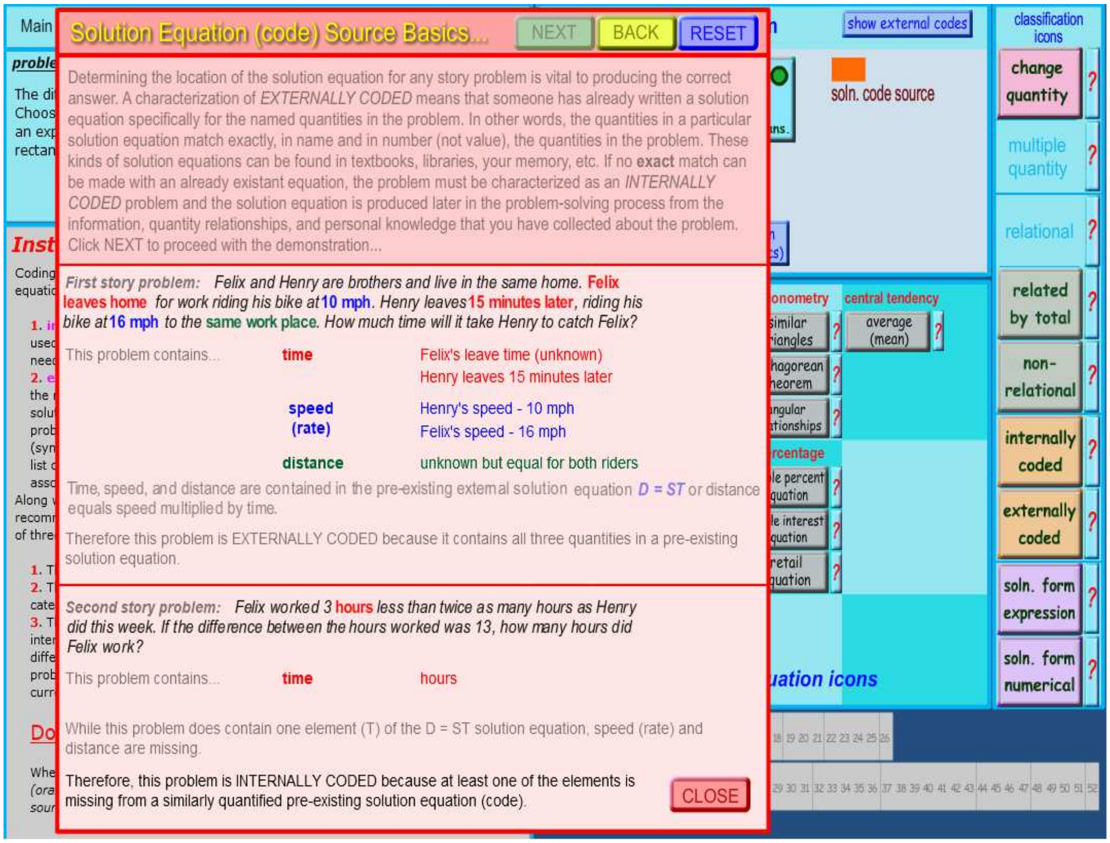
Click **RESET the UTILITY** to start over OR to enable selection of a different mode by closing the instructions presented.

This is a reminder that the utility is in **general operations mode**. Click a **red** button to observe the consequence of your answer to the question in the diamond.

You may also go to the **Main Menu** and change the mode or click on **general instructions** to show them again.

Figure 24. Offering three modes of use (free choice, worked example, do-it-yourself) to teach the decision-making process in determining the second (*quantity relations*) characteristic by answering questions in a specified order, leading to the second characterization. This example shows the student making decisions in response to the question asked in successive diamonds to determine a problem to be relational as shown in the red-ringed circle.

8) Solution Equation Source for determining third characterization graphic – Clicking on the *Solution Equation Source (code basics)* button presents an animated graphic for this determination as shown in Figure 25.



Solution Equation (code) Source Basics...

Determining the location of the solution equation for any story problem is vital to producing the correct answer. A characterization of *EXTERNALLY CODED* means that someone has already written a solution equation specifically for the named quantities in the problem. In other words, the quantities in a particular solution equation match exactly, in name and in number (not value), the quantities in the problem. These kinds of solution equations can be found in textbooks, libraries, your memory, etc. If no **exact** match can be made with an already existant equation, the problem must be characterized as an *INTERNALLY CODED* problem and the solution equation is produced later in the problem-solving process from the information, quantity relationships, and personal knowledge that you have collected about the problem. Click NEXT to proceed with the demonstration...

First story problem: *Felix and Henry are brothers and live in the same home. Felix leaves home for work riding his bike at 10 mph. Henry leaves 15 minutes later, riding his bike at 16 mph to the same work place. How much time will it take Henry to catch Felix?*

This problem contains... **time** Felix's leave time (unknown)
Henry leaves 15 minutes later

speed (rate) Henry's speed - 10 mph
Felix's speed - 16 mph

distance unknown but equal for both riders

Time, speed, and distance are contained in the pre-existing external solution equation *D = ST* or distance equals speed multiplied by time.

Therefore this problem is EXTERNALLY CODED because it contains all three quantities in a pre-existing solution equation.

Second story problem: *Felix worked 3 hours less than twice as many hours as Henry did this week. If the difference between the hours worked was 13, how many hours did Felix work?*

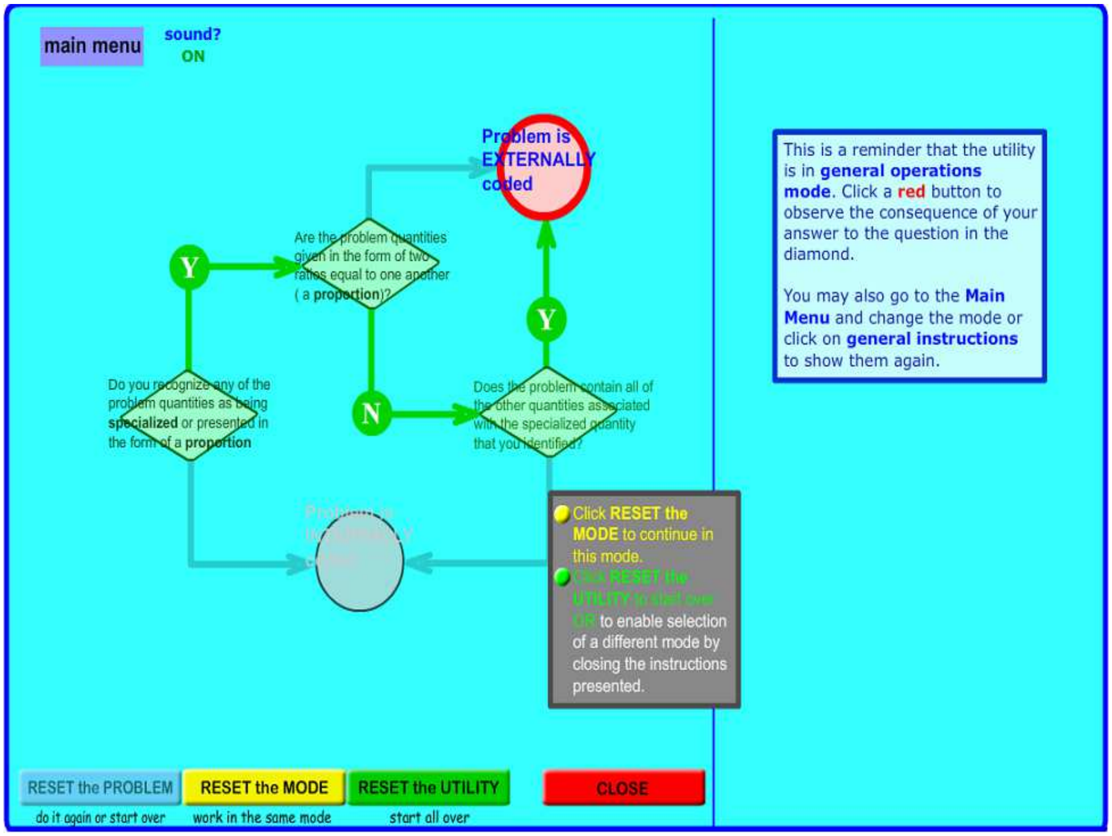
This problem contains... **time** hours

While this problem does contain one element (T) of the D = ST solution equation, speed (rate) and distance are missing.

Therefore, this problem is INTERNALLY CODED because at least one of the elements is missing from a similarly quantified pre-existing solution equation (code).

Figure 25. This graphic animation demonstrates a concrete process that locates certain quantities in the problem that match quantities in pre-existing solution equations that are part of the GED Mathematics Test domain. The first example is a problem whose elements (quantities) match the elements in the pre-existing solution equation, Distance = Rate (speed) · Time (D = RT). Therefore, the problem is characterized as *externally coded*. The second example is a problem that has one element (time) but lacks any of the other elements that a pre-existing solution equation might have. Therefore, the problem is characterized as *internally coded*.

9) The interactive flowchart shown in Figure 26 demonstrates the determination of the third characterization (solution equation source - SES).



Do you recognize any of the problem quantities as being **specialized** or presented in the form of a **proportion**

Are the problem quantities given in the form of two ratios equal to one another (a **proportion**)?

Does the problem contain all of the other quantities associated with the specialized quantity that you identified?

Click **RESET the MODE** to continue in this mode.

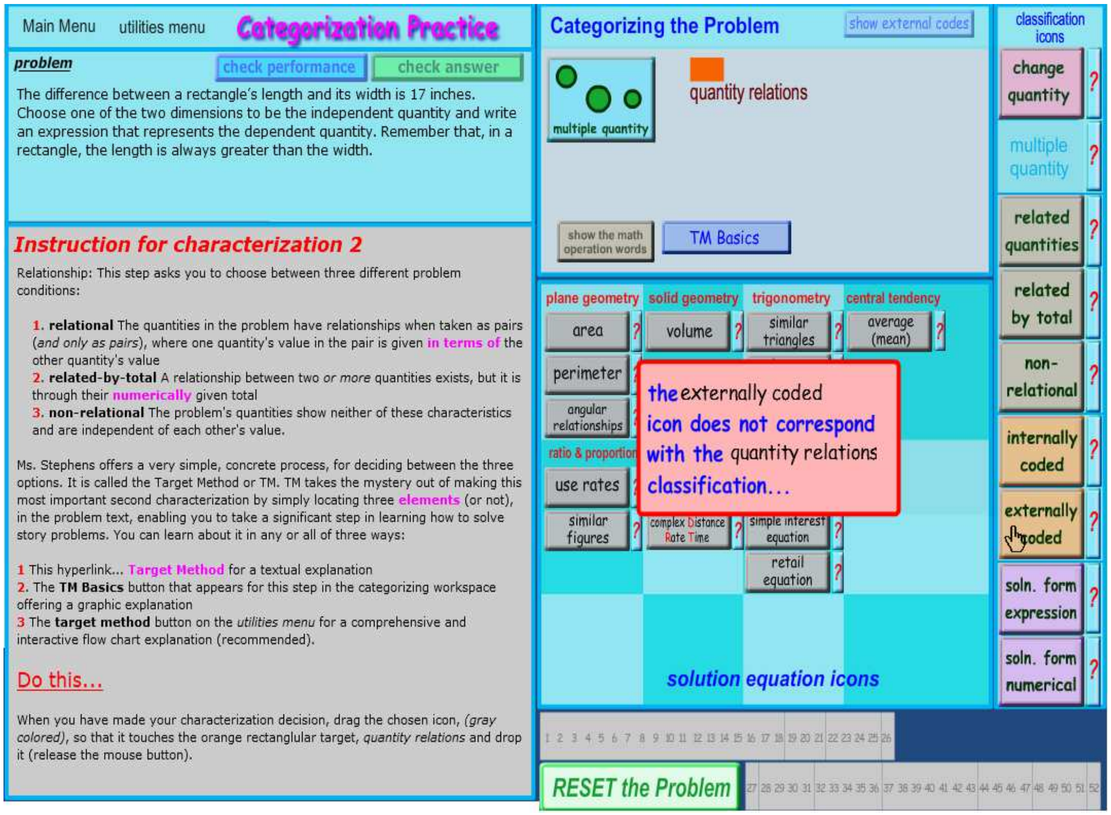
Click **RESET the UTILITY** to start over OR to enable selection of a different mode by closing the instructions presented.

This is a reminder that the utility is in **general operations mode**. Click a **red** button to observe the consequence of your answer to the question in the diamond.

You may also go to the **Main Menu** and change the mode or click on **general instructions** to show them again.

Figure 26. Offering three modes of use (free choice, worked example, do-it-yourself) this interactive flowchart demonstrates a concrete process that calls for the identification of specialized terms, which are names of quantities that occur in pre-existing solution equations (e.g., length, height, supplementary angles, rate of interest, etc.) in a story problem's text. This example shows the student making decisions in response to the question asked in successive diamonds to determine a problem to be externally coded as shown in the red-ringed circle.

10) Choosing an icon from an inappropriate characterization group is shown in Figure 27.



problem

The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quantity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

Instruction for characterization 2

Relationship: This step asks you to choose between three different problem conditions:

1. **relational** The quantities in the problem have relationships when taken as pairs (*and only as pairs*), where one quantity's value in the pair is given **in terms of** the other quantity's value
2. **related-by-total** A relationship between two *or more* quantities exists, but it is through their **numerically** given total
3. **non-relational** The problem's quantities show neither of these characteristics and are independent of each other's value.

Ms. Stephens offers a very simple, concrete process, for deciding between the three options. It is called the Target Method or TM. TM takes the mystery out of making this most important second characterization by simply locating three **elements** (or not), in the problem text, enabling you to take a significant step in learning how to solve story problems. You can learn about it in any or all of three ways:

1 This hyperlink... **Target Method** for a textual explanation
2. The **TM Basics** button that appears for this step in the categorizing workspace offering a graphic explanation
3 The **target method** button on the *utilities menu* for a comprehensive and interactive flow chart explanation (recommended).

Do this...

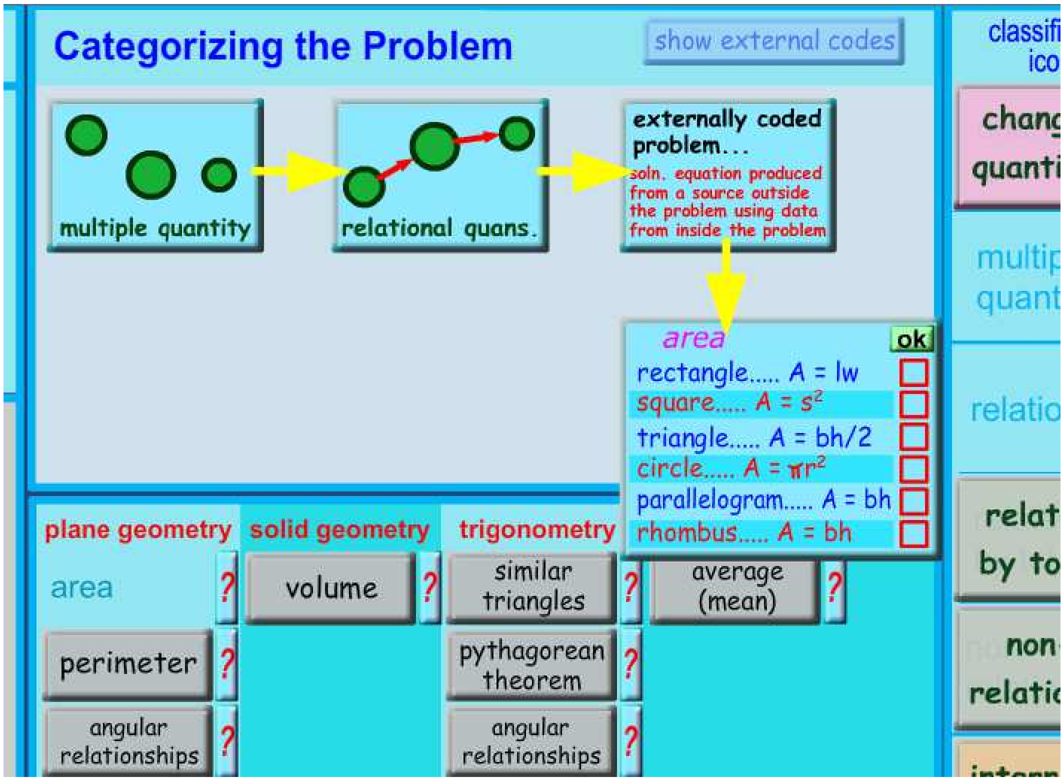
When you have made your characterization decision, drag the chosen icon, *(gray colored)*, so that it touches the orange rectanglular target, *quantity relations* and drop it (release the mouse button).

the externally coded icon does not correspond with the quantity relations classification...

Figure 27. Characteristic icon groupings are the same color. While all but the appropriate icons are disabled for drag and drop articulation, as a teachable moment, an error message regarding the misapplication of a particular characteristic icon appears then moments later, floats off the workspace.

11) Please refer to Figure 27 above and in the top left area of the Work Space. When dropped on a target each icon expands to deploy as a graphic representation of the characteristic. Here, the multiple quantity characteristic is represented by three different sized circles that represent the three distinct quantities (length, width, and difference) in the story problem. Such an action follows Marshall's principle that "visual diagrams influence conceptual development by functioning as an anchor for the student's models when used to represent problem structure as simply and uniquely as possible." (1995).

12) Solution equation icons deploy as graphic representations when dropped on a target as shown in Figure 28, below.

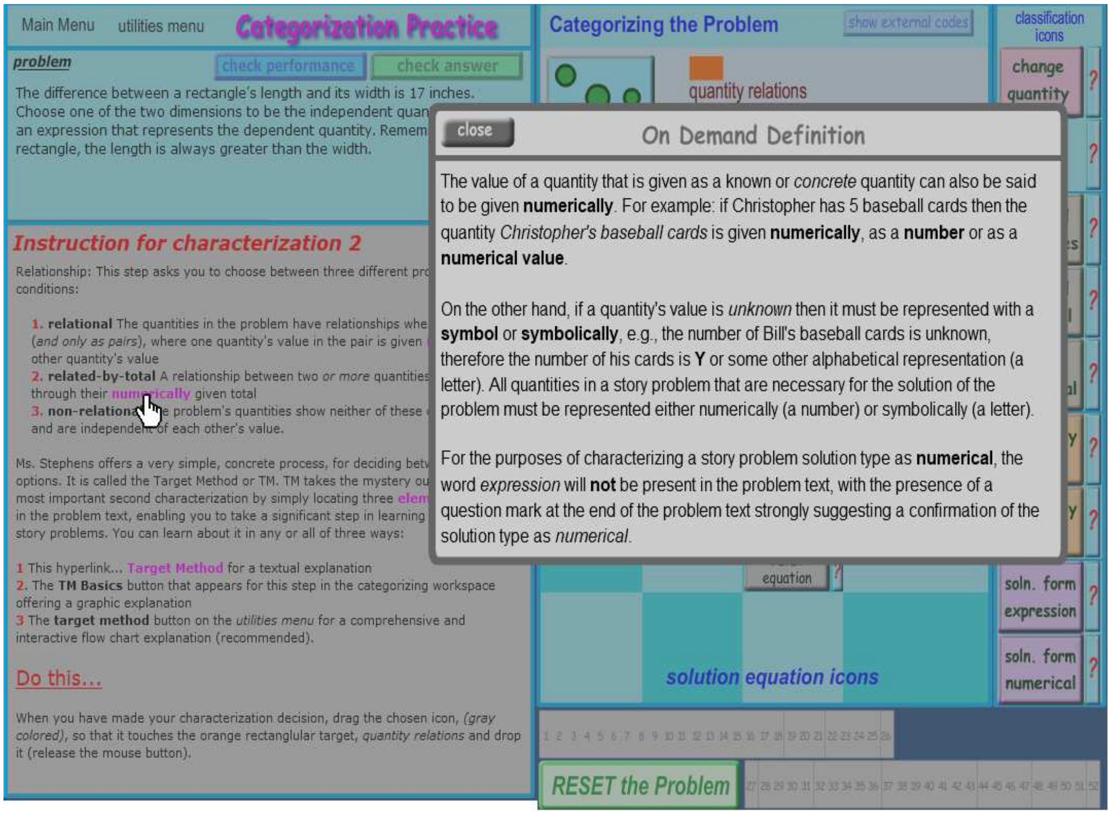


soln. equation produced from a source outside the problem using data from inside the problem

rectangle..... $A = lw$ □
square..... $A = s^2$ □
triangle..... $A = bh/2$ □
circle..... $A = \pi r^2$ □
parallelogram..... $A = bh$ □
rhombus..... $A = bh$ □

13) *Figure 28*. Similar to characteristic icons except that the solution equation(s) associated with a particular problem situation are shown. In this case, all solution equations that are associated with **area** are shown. Not all external solution equation for a particular problem situation are multiple. For example, the retail equation icon shows a singular **Selling Price = Cost + Profit**.

14) Hyperlinked text is shown in Figure 29, below.



problem

The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quan an expression that represents the dependent quantity. Remem rectangle, the length is always greater than the width.

Instruction for characterization 2

Relationship: This step asks you to choose between three different pro conditions:

1. **relational** The quantities in the problem have relationships whe (*and only as pairs*), where one quantity's value in the pair is given other quantity's value
2. **related-by-total** A relationship between two *or more* quantities through their **numerically** given total
3. **non-relationa** e problem's quantities show neither of these and are independent of each other's value.

Ms. Stephens offers a very simple, concrete process, for deciding betw options. It is called the Target Method or TM. TM takes the mystery ou most important second characterization by simply locating three **elem** in the problem text, enabling you to take a significant step in learning story problems. You can learn about it in any or all of three ways:

1 This hyperlink... **Target Method** for a textual explanation
2. The **TM Basics** button that appears for this step in the categorizing workspace offering a graphic explanation
3 The **target method** button on the *utilities menu* for a comprehensive and interactive flow chart explanation (recommended).

Do this...

When you have made your characterization decision, drag the chosen icon, *(gray colored)*, so that it touches the orange rectanglular target, *quantity relations* and drop it (release the mouse button).

On Demand Definition

The value of a quantity that is given as a known or *concrete* quantity can also be said to be given **numerically**. For example: if Christopher has 5 baseball cards then the quantity *Christopher's baseball cards* is given **numerically**, as a **number** or as a **numerical value**.

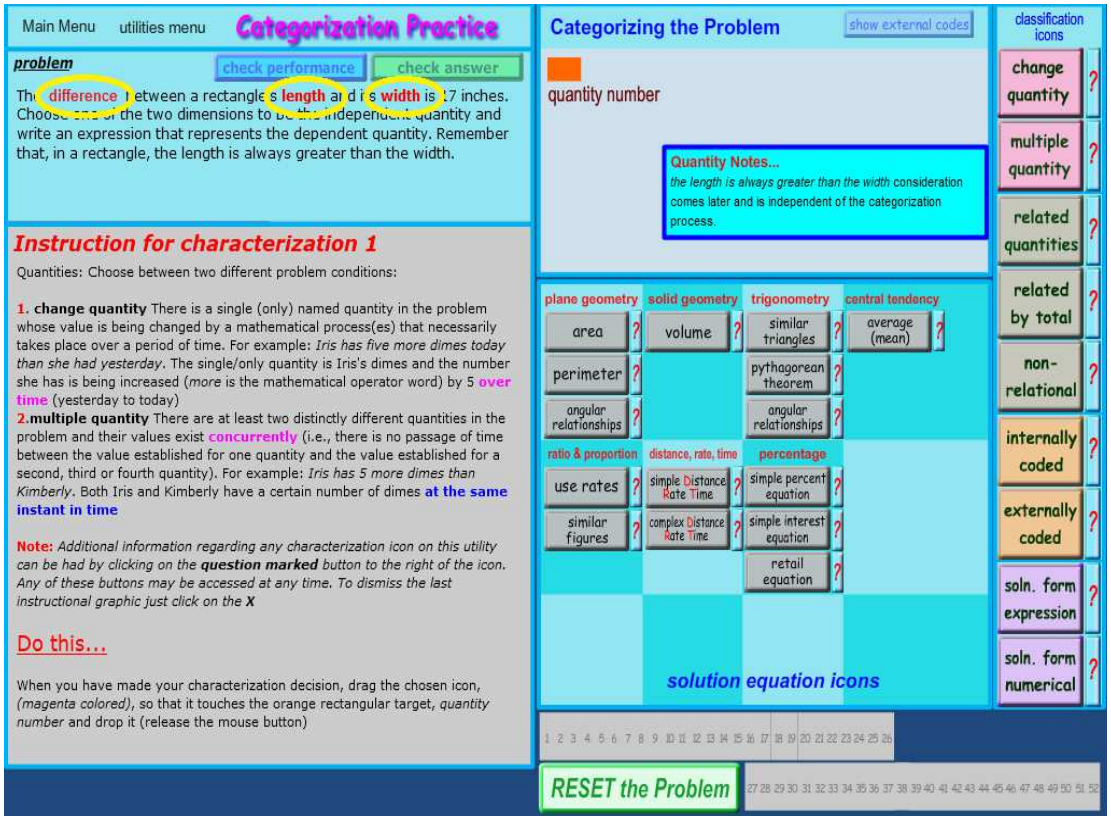
On the other hand, if a quantity's value is *unknown* then it must be represented with a **symbol** or **symbolically**, e.g., the number of Bill's baseball cards is unknown, therefore the number of his cards is **Y** or some other alphabetical representation (a letter). All quantities in a story problem that are necessary for the solution of the problem must be represented either numerically (a number) or symbolically (a letter).

For the purposes of characterizing a story problem solution type as **numerical**, the word *expression* will **not** be present in the problem text, with the presence of a question mark at the end of the problem text strongly suggesting a confirmation of the solution type as *numerical*.

Figure 29. Terms or phrases that are bold and fuchsia-colored (e.g., the term in the graphic above, *numerically*), display textual or graphical representations of a term, phrase, method, or process when clicked by the user, serving as a just-in-time avenue for effective and efficient learning. When the item is on-screen, the utility is disabled.

15) Characterization decisions are dependent on the words that make up the problem text from both a literal perspective and an intuitive sense developed from problem-solving and life experiences. During the observational sessions it became apparent that an unanticipated level of obstacle (below concrete procedural) was being encountered by the subjects as they struggled with identifying the words critical to a particular characterization. An additional level of scaffolding had to be constructed to support the algorithmic processes introduced into the problem-solving process by the MS-CPU as novices had to begin the growth of problem-solving skills on a largely literal basis. For this reason the MS-CPU allows the

problem-solver to expose the terms in a story problem that are germane to any of the first (named quantities), second (all terms associated with the Target Method), third (specialized terms found in pre-existing solution equations), and fifth (terms that key the form or type of solution called for) characterizations at the click of a button. Button placement is always in the left-bottom corner of the Work Space*.* Words and phrases germane to a particular characterization decision can be highlighted for any problem as shown in Figures 30 - 33.



problem

The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quantity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

Instruction for characterization 1

Quantities: Choose between two different problem conditions:

1. **change quantity** There is a single (only) named quantity in the problem whose value is being changed by a mathematical process(es) that necessarily takes place over a period of time. For example: *Iris has five more dimes today than she had yesterday*. The single/only quantity is Iris's dimes and the number she has is being increased (*more* is the mathematical operator word) by 5 **over time** (yesterday to today)
2.**multiple quantity** There are at least two distinctly different quantities in the problem and their values exist **concurrently** (i.e., there is no passage of time between the value established for one quantity and the value established for a second, third or fourth quantity). For example: *Iris has 5 more dimes than Kimberly*. Both Iris and Kimberly have a certain number of dimes **at the same instant in time**

Note: *Additional information regarding any characterization icon on this utility can be had by clicking on the* ***question marked*** *button to the right of the icon. Any of these buttons may be accessed at any time. To dismiss the last instructional graphic just click on the* ***X***

Do this...

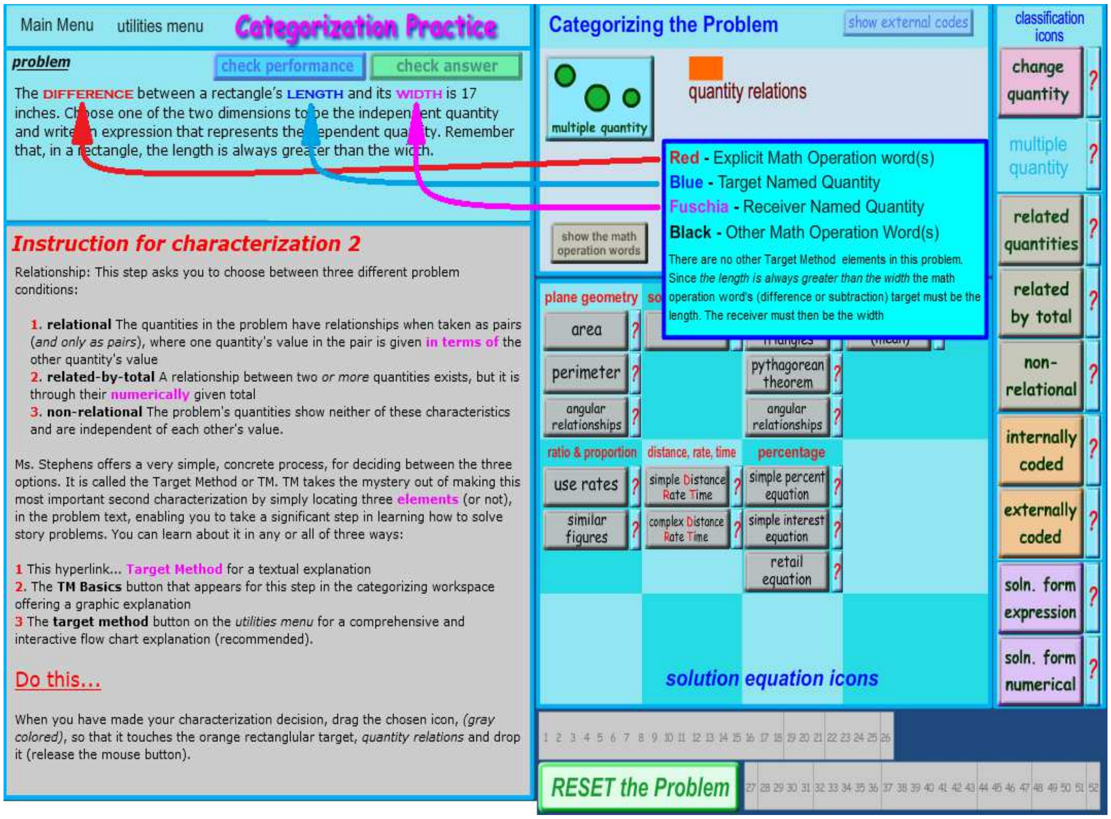
When you have made your characterization decision, drag the chosen icon, *(magenta colored)*, so that it touches the orange rectangular target, *quantity number* and drop it (release the mouse button)

Quantity Notes...

the length is always greater than the width consideration comes later and is independent of the categorization process.

Figure 30. Clicking the **show the named quantities** button highlights (in red) the named quantities that appear in the problem text. The ability to identify these quantities are critical to the user in determining whether the story problem has multiple discrete quantities with a value (either known or unknown) for each quantity OR a single quantity whose value has changed over time (a beginning state and an end state). For example, this story problem contains three quantities (length, width, and difference). In the context of the problem, the length is Y, the width is W, and the difference is 17, so the problem is characterized as

multiple quantity. This feature has limits on its use as noted in the Discussion chapter (pg. 124 – final paragraph).



The DIFFERENCE between a rectangle's LENGTH and its WIDTH is 17 inches. Choose one of the two dimensions to be the independent quantity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

Red - Explicit Math Operation word(s)
Blue - Target Named Quantity
Fuschia - Receiver Named Quantity
Black - Other Math Operation Word(s)

There are no other Target Method elements in this problem. Since *the length is always greater than the width* the math operation word's (difference or subtraction) target must be the length. The receiver must then be the width

Instruction for characterization 2

Relationship: This step asks you to choose between three different problem conditions:

1. **relational** The quantities in the problem have relationships when taken as pairs (*and only as pairs*), where one quantity's value in the pair is given **in terms of** the other quantity's value
2. **related-by-total** A relationship between two *or more* quantities exists, but it is through their **numerically** given total
3. **non-relational** The problem's quantities show neither of these characteristics and are independent of each other's value.

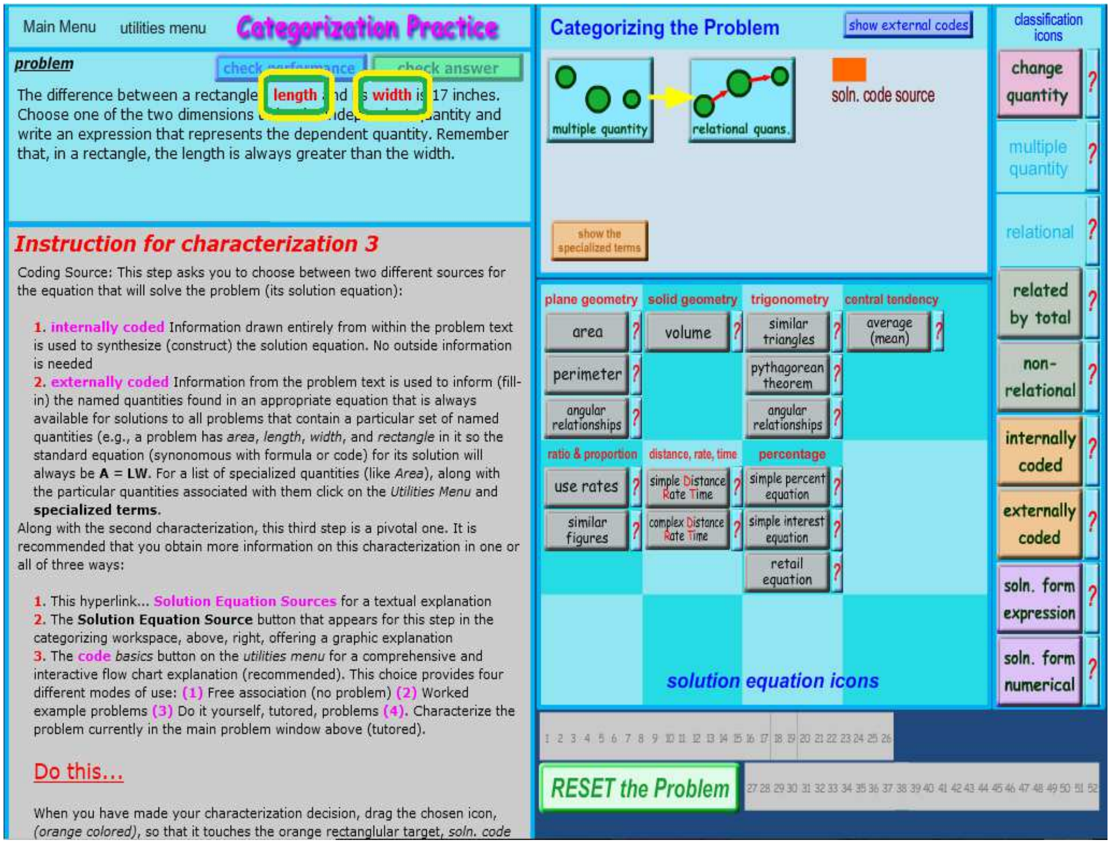
Ms. Stephens offers a very simple, concrete process, for deciding between the three options. It is called the Target Method or TM. TM takes the mystery out of making this most important second characterization by simply locating three **elements** (or not), in the problem text, enabling you to take a significant step in learning how to solve story problems. You can learn about it in any or all of three ways:

1 This hyperlink... **Target Method** for a textual explanation
2. The **TM Basics** button that appears for this step in the categorizing workspace offering a graphic explanation
3 The **target method** button on the *utilities menu* for a comprehensive and interactive flow chart explanation (recommended).

Do this...

When you have made your characterization decision, drag the chosen icon, *(gray colored)*, so that it touches the orange rectanglular target, *quantity relations* and drop it (release the mouse button).

Figure 31. Clicking the **show the math operation words** button highlights (in red) the word(s) that indicate a mathematical operation; also shown are the two other elements that the Target Method identifies: (1) the target quantity (in blue) and (2) the receiver quantity (in fuchsia). The legend in the Work Space links color to the element type. This feature has limits on its use as noted in the Discussion chapter (pg. 124 – final paragraph).



problem

The difference between a rectangle length and s width is 17 inches. Choose one of the two dimensions ide antity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

Instruction for characterization 3

Coding Source: This step asks you to choose between two different sources for the equation that will solve the problem (its solution equation):

1. internally coded Information drawn entirely from within the problem text is used to synthesize (construct) the solution equation. No outside information is needed
2. externally coded Information from the problem text is used to inform (fill-in) the named quantities found in an appropriate equation that is always available for solutions to all problems that contain a particular set of named quantities (e.g., a problem has *area*, *length*, *width*, and *rectangle* in it so the standard equation (synonomous with formula or code) for its solution will always be **A = LW**. For a list of specialized quantities (like *Area*), along with the particular quantities associated with them click on the *Utilities Menu* and **specialized terms**.

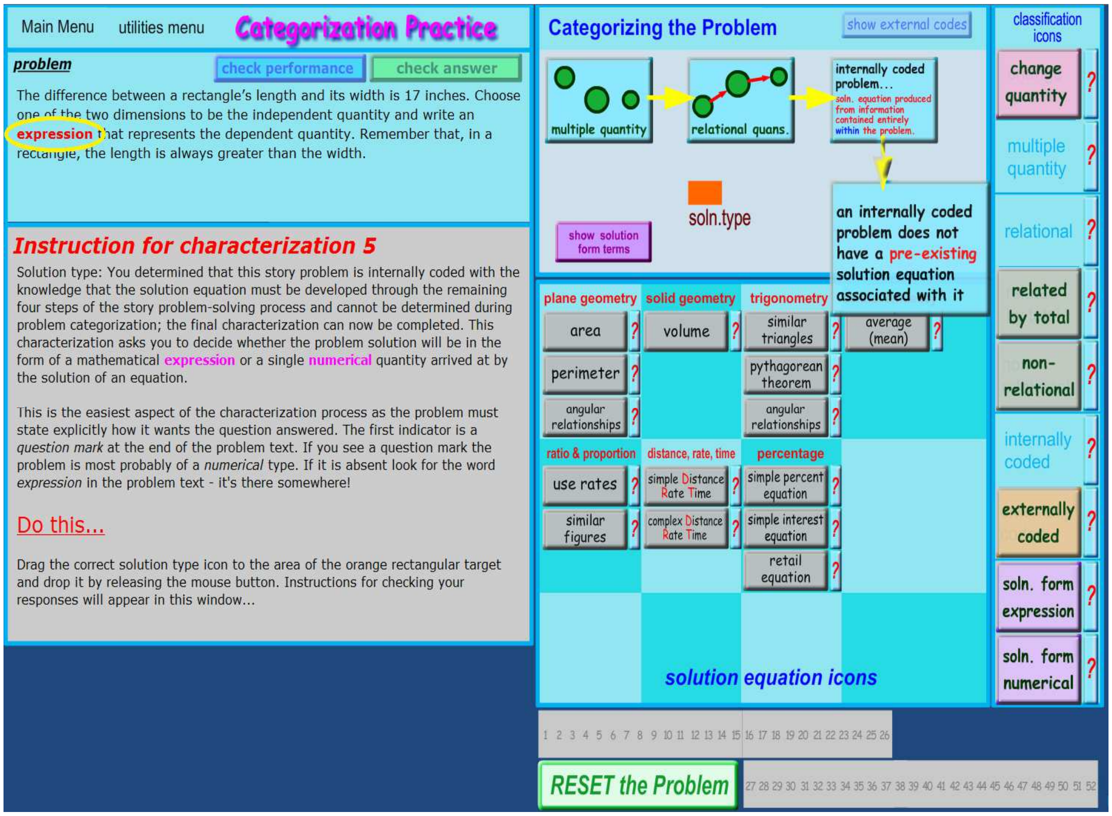
Along with the second characterization, this third step is a pivotal one. It is recommended that you obtain more information on this characterization in one or all of three ways:

1. This hyperlink... Solution Equation Sources for a textual explanation
2. The **Solution Equation Source** button that appears for this step in the categorizing workspace, above, right, offering a graphic explanation
3. The code *basics* button on the *utilities menu* for a comprehensive and interactive flow chart explanation (recommended). This choice provides four different modes of use: (1) Free association (no problem) (2) Worked example problems (3) Do it yourself, tutored, problems (4). Characterize the problem currently in the main problem window above (tutored).

Do this...

When you have made your characterization decision, drag the chosen icon, *(orange colored)*, so that it touches the orange rectanglular target, *soln. code*

Figure 32. Clicking the **show the specialized terms** button highlights (in red) the word(s) that are associated with pre-existing (external) solution equations. It is the responsibility of the user to make sure that the number and kind (name) of the quantities in the problem match the chosen solution equation. The directory of specialized terms (Figure 41, pg. 80) provides associated terms when one specialized term has been identified. For example, this problem has two specialized terms (length and width), so it could be associated with either the perimeter or area of a rectangle. However, absent is the quantity area (or perimeter) and the descriptor, rectangle, so this must be characterized as an *internally coded* problem. This feature has limits on its use as noted in the Discussion chapter (pg. 124 – final paragraph).



The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quantity and write an **expression** that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

Instruction for characterization 5

Solution type: You determined that this story problem is internally coded with the knowledge that the solution equation must be developed through the remaining four steps of the story problem-solving process and cannot be determined during problem categorization; the final characterization can now be completed. This characterization asks you to decide whether the problem solution will be in the form of a mathematical **expression** or a single **numerical** quantity arrived at by the solution of an equation.

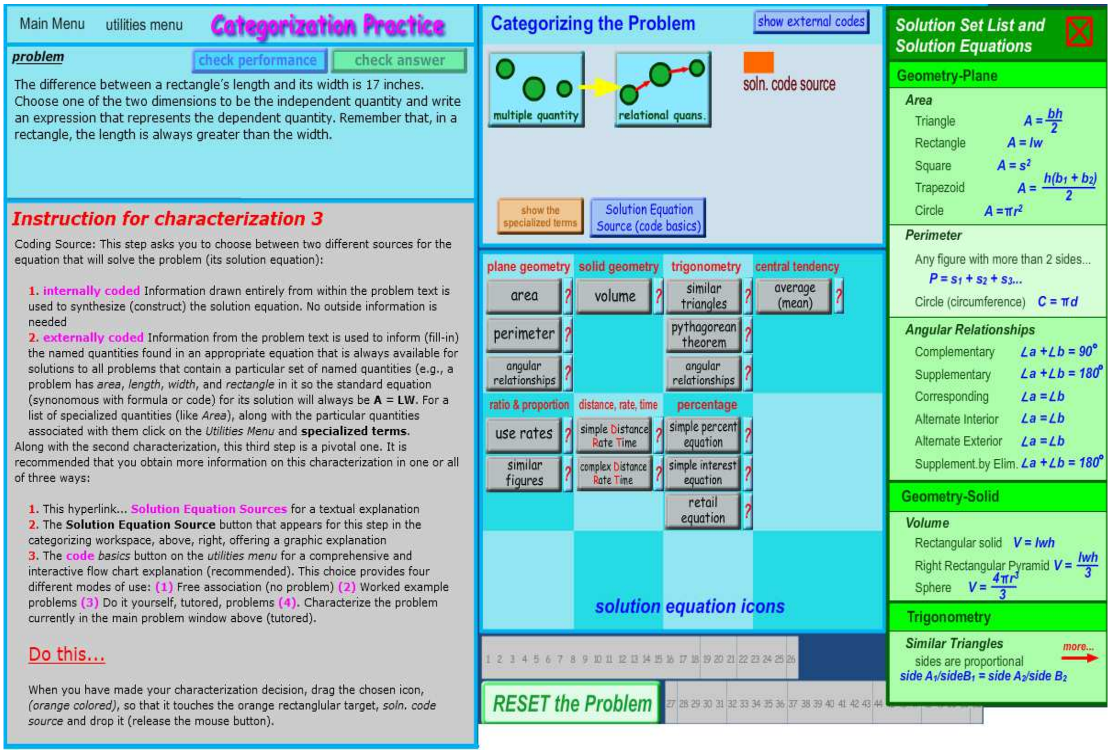
This is the easiest aspect of the characterization process as the problem must state explicitly how it wants the question answered. The first indicator is a *question mark* at the end of the problem text. If you see a question mark the problem is most probably of a *numerical* type. If it is absent look for the word *expression* in the problem text - it's there somewhere!

Do this...

Drag the correct solution type icon to the area of the orange rectangular target and drop it by releasing the mouse button. Instructions for checking your responses will appear in this window...

16) *Figure 33.* Clicking the **show solution form terms** button highlights (in red) the word(s) that are associated with the current problem's solution type (*expression* or *numerical*). If the word *expression* is contained in the problem text, the solution type is, without doubt, an expression. If the problem ends with a question mark, the solution type is most probably *numerical*, but a check of the problem text for the absence of the word expression needs to be made to confirm that the characterization is numerical. This feature has limits on its use as noted in the Discussion chapter (pg. 124 – final paragraph).

17) Graphic presentations of every pre-existing (external) solution equation in the GED Mathematics Test domain are available as shown in Figure 34.



problem

The difference between a rectangle's length and its width is 17 inches. Choose one of the two dimensions to be the independent quantity and write an expression that represents the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.

Instruction for characterization 3

Coding Source: This step asks you to choose between two different sources for the equation that will solve the problem (its solution equation):

1. **internally coded** Information drawn entirely from within the problem text is used to synthesize (construct) the solution equation. No outside information is needed
2. **externally coded** Information from the problem text is used to inform (fill-in) the named quantities found in an appropriate equation that is always available for solutions to all problems that contain a particular set of named quantities (e.g., a problem has *area*, *length*, *width*, and *rectangle* in it so the standard equation (synonomous with formula or code) for its solution will always be **A = LW**. For a list of specialized quantities (like *Area*), along with the particular quantities associated with them click on the *Utilities Menu* and **specialized terms**.

Along with the second characterization, this third step is a pivotal one. It is recommended that you obtain more information on this characterization in one or all of three ways:

1. This hyperlink... **Solution Equation Sources** for a textual explanation
2. The **Solution Equation Source** button that appears for this step in the categorizing workspace, above, right, offering a graphic explanation
3. The **code** *basics* button on the *utilities menu* for a comprehensive and interactive flow chart explanation (recommended). This choice provides four different modes of use: (1) Free association (no problem) (2) Worked example problems (3) Do it yourself, tutored, problems (4). Characterize the problem currently in the main problem window above (tutored).

Do this...

When you have made your characterization decision, drag the chosen icon, *(orange colored)*, so that it touches the orange rectanglular target, *soln. code source* and drop it (release the mouse button).

Solution Set List and Solution Equations

Geometry-Plane

Area

Triangle $A = \frac{bh}{2}$
Rectangle $A = lw$
Square $A = s^2$
Trapezoid $A = \frac{h(b_1 + b_2)}{2}$
Circle $A = \pi r^2$

Perimeter

Any figure with more than 2 sides...
$P = s_1 + s_2 + s_3...$
Circle (circumference) $C = \pi d$

Angular Relationships

Complementary $\angle a + \angle b = 90°$
Supplementary $\angle a + \angle b = 180°$
Corresponding $\angle a = \angle b$
Alternate Interior $\angle a = \angle b$
Alternate Exterior $\angle a = \angle b$
Supplement.by Elim. $\angle a + \angle b = 180°$

Geometry-Solid

Volume

Rectangular solid $V = lwh$
Right Rectangular Pyramid $V = \frac{lwh}{3}$
Sphere $V = \frac{4\pi r^3}{3}$

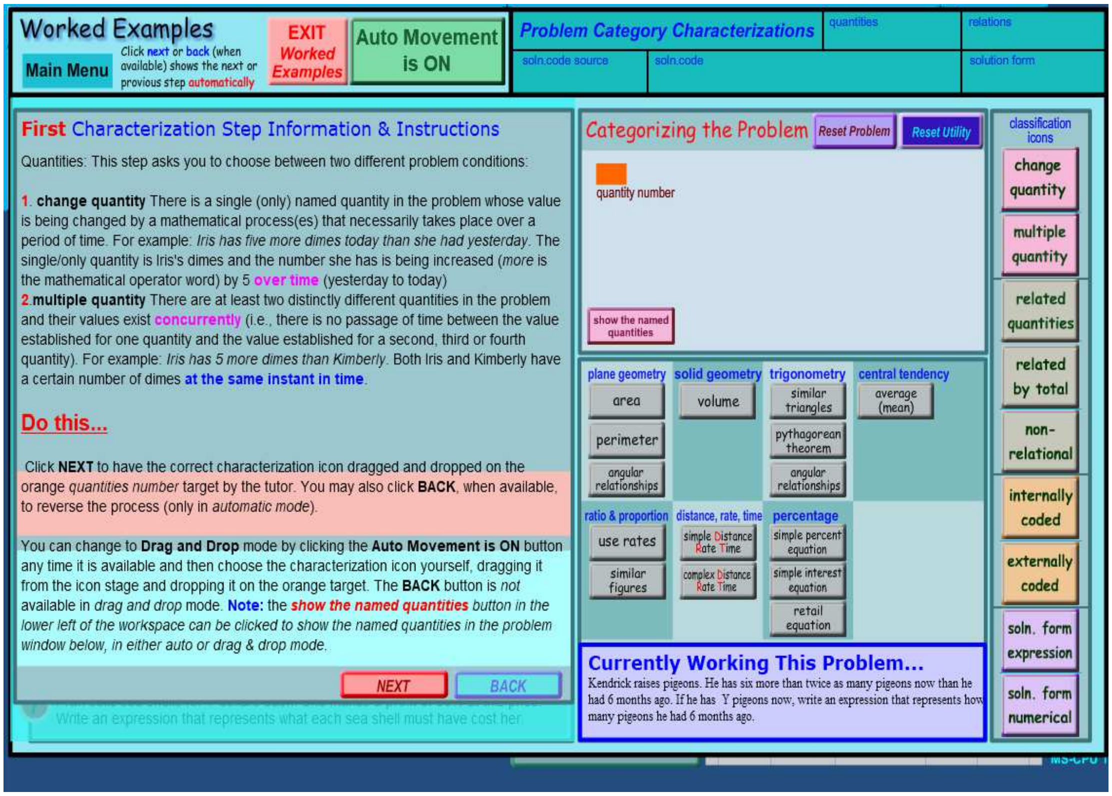
Trigonometry

Similar Triangles

sides are proportional
side A_1/sideB_1 = side A_2/side B_2

Figure 34. Fundamental to the categorization of story problems is the possession, in long term memory, of the various external solution equations that a GED student (or any story problem-solver) will find necessary to solve a problem that is externally coded. Clicking the **show external codes** button on the right side of the MS-CPU, above the Work Space displays a two-page graphic with all of the pre-existing solution equations (codes) with which the GED student must be familiar. The question-marked areas in the solution equation section of the Work Space produce similar graphics, but only of a particular family or set. Presenting the codes all at once may produce some additional learning benefit as the student reads even those that are not of interest until the search concludes.

18) Of critical service in learning to categorize story problems is the Worked Examples (WE) utility shown in Figures 35 – 39 (**Auto Movement is On** mode).



First Characterization Step Information & Instructions

Quantities: This step asks you to choose between two different problem conditions:

1. **change quantity** There is a single (only) named quantity in the problem whose value is being changed by a mathematical process(es) that necessarily takes place over a period of time. For example: *Iris has five more dimes today than she had yesterday*. The single/only quantity is Iris's dimes and the number she has is being increased (*more* is the mathematical operator word) by 5 **over time** (yesterday to today)
2.**multiple quantity** There are at least two distinctly different quantities in the problem and their values exist **concurrently** (i.e., there is no passage of time between the value established for one quantity and the value established for a second, third or fourth quantity). For example: *Iris has 5 more dimes than Kimberly*. Both Iris and Kimberly have a certain number of dimes **at the same instant in time**.

Do this...

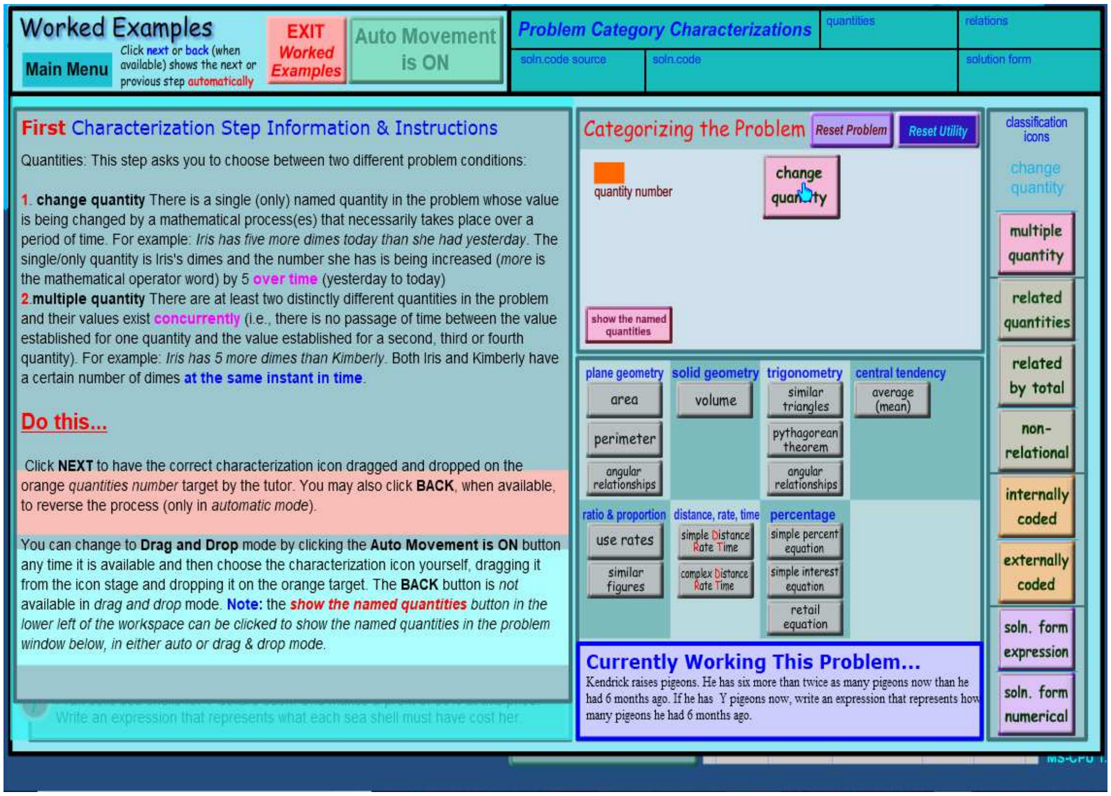
Click **NEXT** to have the correct characterization icon dragged and dropped on the orange *quantities number* target by the tutor. You may also click **BACK**, when available, to reverse the process (only in *automatic mode*).

You can change to **Drag and Drop** mode by clicking the **Auto Movement is ON** button any time it is available and then choose the characterization icon yourself, dragging it from the icon stage and dropping it on the orange target. The **BACK** button is *not* available in *drag and drop* mode. **Note:** the ***show the named quantities*** *button in the lower left of the workspace can be clicked to show the named quantities in the problem window below, in either auto or drag & drop mode.*

Currently Working This Problem...

Kendrick raises pigeons. He has six more than twice as many pigeons now than he had 6 months ago. If he has Y pigeons now, write an expression that represents how many pigeons he had 6 months ago.

Figure 35. Arguably the strongest method for effective and efficient delivery of pedagogy is worked examples. In this example, the MS-CPU worked examples utility (WEU) is running in the Automatic Mode that demonstrates each step for each of the five characterizations at the click of the **NEXT** button. An example problem has been selected and appears below the Solution Equations staging area. The first characterization instructions appear in the instructions window, waiting for the user to read them, and then to click the NEXT button, the result of which is shown in Figure 36. Note that the green button, center top, shows Auto Movement is ON. The WEU user interface is virtually identical to the MS-CPU.



First Characterization Step Information & Instructions

Quantities: This step asks you to choose between two different problem conditions:

1. **change quantity** There is a single (only) named quantity in the problem whose value is being changed by a mathematical process(es) that necessarily takes place over a period of time. For example: *Iris has five more dimes today than she had yesterday*. The single/only quantity is Iris's dimes and the number she has is being increased (*more* is the mathematical operator word) by 5 **over time** (yesterday to today)

2. **multiple quantity** There are at least two distinctly different quantities in the problem and their values exist **concurrently** (i.e., there is no passage of time between the value established for one quantity and the value established for a second, third or fourth quantity). For example: *Iris has 5 more dimes than Kimberly*. Both Iris and Kimberly have a certain number of dimes **at the same instant in time**.

Do this...

Click **NEXT** to have the correct characterization icon dragged and dropped on the orange *quantities number* target by the tutor. You may also click **BACK**, when available, to reverse the process (only in *automatic mode*).

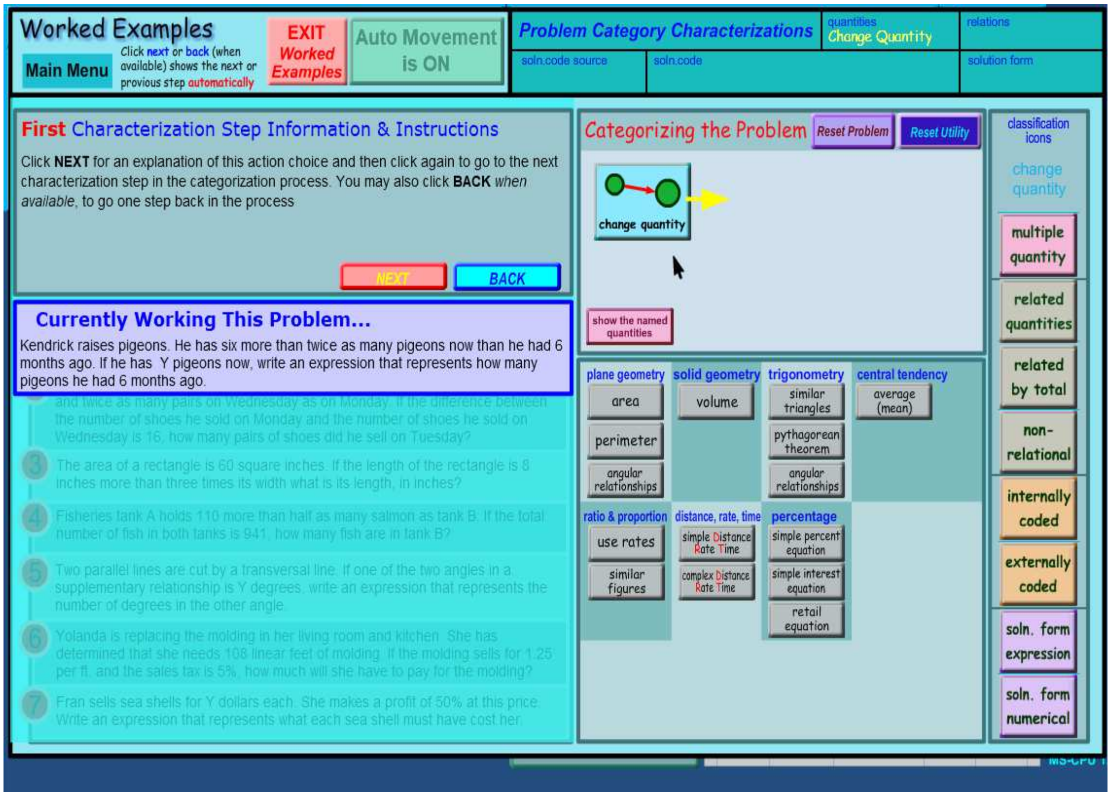
You can change to **Drag and Drop** mode by clicking the **Auto Movement is ON** button any time it is available and then choose the characterization icon yourself, dragging it from the icon stage and dropping it on the orange target. The **BACK** button is *not* available in *drag and drop* mode. **Note:** the ***show the named quantities*** *button in the lower left of the workspace can be clicked to show the named quantities in the problem window below, in either auto or drag & drop mode.*

Write an expression that represents what each sea shell must have cost her.

Currently Working This Problem...

Kendrick raises pigeons. He has six more than twice as many pigeons now than he had 6 months ago. If he has Y pigeons now, write an expression that represents how many pigeons he had 6 months ago.

Figure 36. In this example, the MS-CPU Worked Examples (WE) utility has been advanced by clicking the **NEXT** button. The animation begins with the arrow cursor moving to the *change quantity* icon, the cursor changes to a hand and begins to move the icon to the first target, as the user would do in the MS-CPU. Note the Problem Category Characterizations in the upper right margin of the MS-CPU for posting each correct characterization.



First Characterization Step Information & Instructions

Click **NEXT** for an explanation of this action choice and then click again to go to the next characterization step in the categorization process. You may also click **BACK** *when available*, to go one step back in the process

Currently Working This Problem...

Kendrick raises pigeons. He has six more than twice as many pigeons now than he had 6 months ago. If he has Y pigeons now, write an expression that represents how many pigeons he had 6 months ago.

and twice as many pairs on Wednesday as on Monday. If the difference between the number of shoes he sold on Monday and the number of shoes he sold on Wednesday is 16, how many pairs of shoes did he sell on Tuesday?

3 The area of a rectangle is 60 square inches. If the length of the rectangle is 8 inches more than three times its width what is its length, in inches?

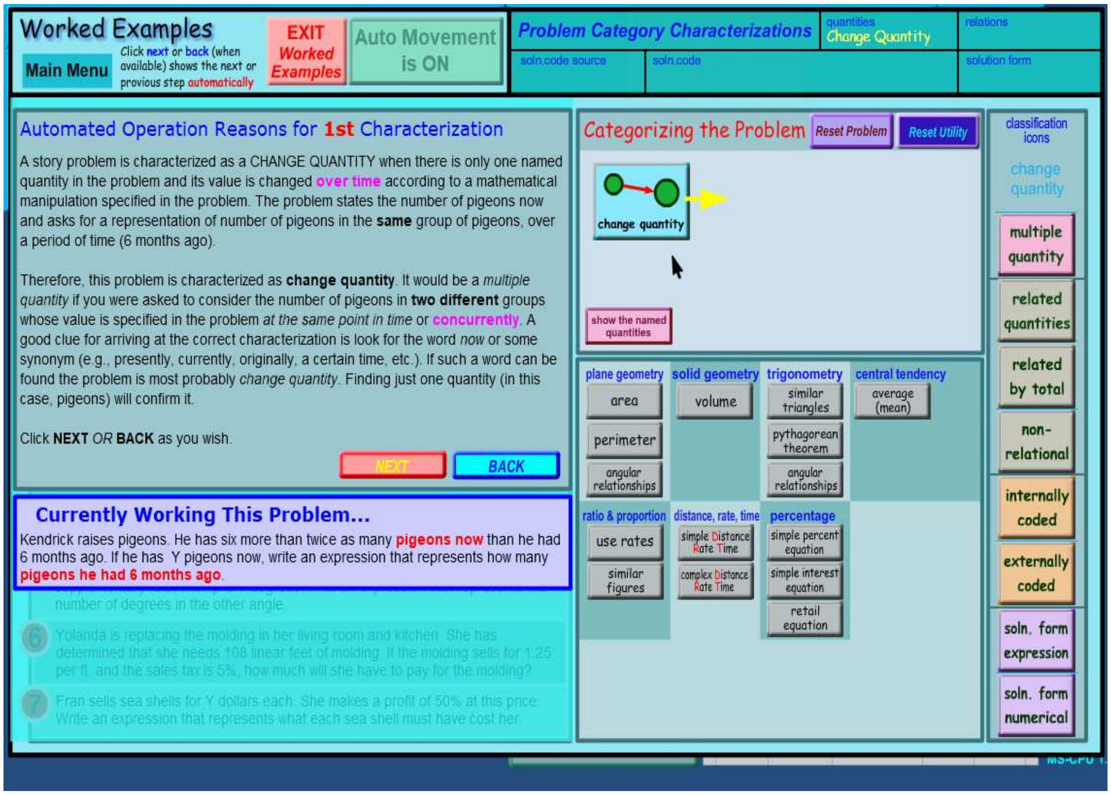
4 Fisheries tank A holds 110 more than half as many salmon as tank B. If the total number of fish in both tanks is 941, how many fish are in tank B?

5 Two parallel lines are cut by a transversal line. If one of the two angles in a supplementary relationship is Y degrees, write an expression that represents the number of degrees in the other angle.

6 Yolanda is replacing the molding in her living room and kitchen. She has determined that she needs 108 linear feet of molding. If the molding sells for 1.25 per ft. and the sales tax is 5%, how much will she have to pay for the molding?

7 Fran sells sea shells for Y dollars each. She makes a profit of 50% at this price. Write an expression that represents what each sea shell must have cost her.

Figure 37. The animation is complete with the change quantity icon in place and the characteristic posted. The current problem window is under the instructions window as usual unless the instruction window is too large (as it was in the previous step), in which case it goes to the side. The system is waiting for the user to click the **NEXT** button which produces the user interface shown in Figure 38.



Automated Operation Reasons for 1st Characterization

A story problem is characterized as a CHANGE QUANTITY when there is only one named quantity in the problem and its value is changed over time according to a mathematical manipulation specified in the problem. The problem states the number of pigeons now and asks for a representation of number of pigeons in the **same** group of pigeons, over a period of time (6 months ago).

Therefore, this problem is characterized as **change quantity**. It would be a *multiple quantity* if you were asked to consider the number of pigeons in **two different** groups whose value is specified in the problem *at the same point in time* or **concurrently**. A good clue for arriving at the correct characterization is look for the word *now* or some synonym (e.g., presently, currently, originally, a certain time, etc.). If such a word can be found the problem is most probably *change quantity*. Finding just one quantity (in this case, pigeons) will confirm it.

Click **NEXT** *OR* **BACK** as you wish.

Currently Working This Problem...

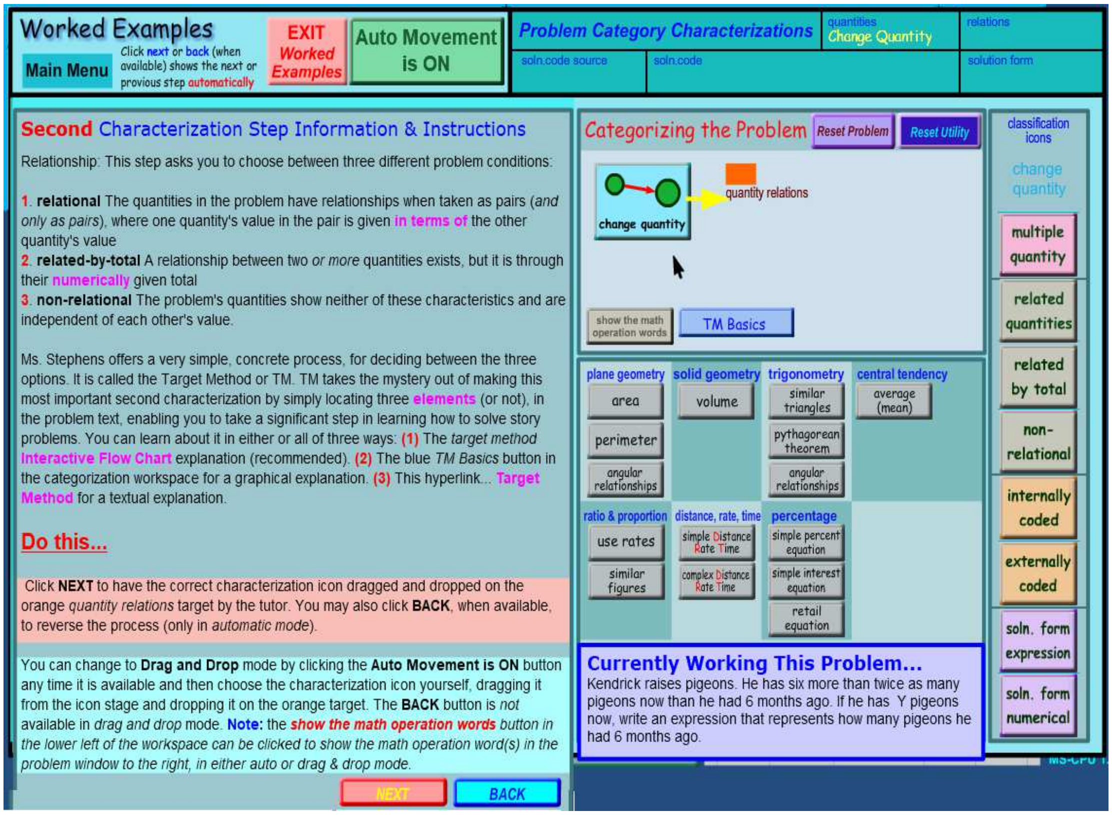
Kendrick raises pigeons. He has six more than twice as many **pigeons now** than he had 6 months ago. If he has Y pigeons now, write an expression that represents how many **pigeons he had 6 months ago**.

number of degrees in the other angle.

6 Yolanda is replacing the molding in her living room and kitchen. She has determined that she needs 108 linear feet of molding. If the molding sells for 1.25 per ft. and the sales tax is 5%, how much will she have to pay for the molding?

7 Fran sells sea shells for Y dollars each. She makes a profit of 50% at this price. Write an expression that represents what each sea shell must have cost her.

Figure 38. The rationale for the first characterization shows in the instructions window and the quantity names are highlighted automatically in the problem window. The system is waiting for the user to click the **NEXT** button which produces beginning step of the next characterization (*quantity relations*) shown in Figure 39.



Second Characterization Step Information & Instructions

Relationship: This step asks you to choose between three different problem conditions:

1. **relational** The quantities in the problem have relationships when taken as pairs (*and only as pairs*), where one quantity's value in the pair is given **in terms of** the other quantity's value
2. **related-by-total** A relationship between two *or more* quantities exists, but it is through their **numerically** given total
3. **non-relational** The problem's quantities show neither of these characteristics and are independent of each other's value.

Ms. Stephens offers a very simple, concrete process, for deciding between the three options. It is called the Target Method or TM. TM takes the mystery out of making this most important second characterization by simply locating three **elements** (or not), in the problem text, enabling you to take a significant step in learning how to solve story problems. You can learn about it in either or all of three ways: **(1)** The *target method* **Interactive Flow Chart** explanation (recommended). **(2)** The blue *TM Basics* button in the categorization workspace for a graphical explanation. **(3)** This hyperlink... **Target Method** for a textual explanation.

Do this...

Click **NEXT** to have the correct characterization icon dragged and dropped on the orange *quantity relations* target by the tutor. You may also click **BACK**, when available, to reverse the process (only in *automatic mode*).

You can change to **Drag and Drop** mode by clicking the **Auto Movement is ON** button any time it is available and then choose the characterization icon yourself, dragging it from the icon stage and dropping it on the orange target. The **BACK** button is *not* available in *drag and drop* mode. **Note:** the ***show the math operation words*** *button in the lower left of the workspace can be clicked to show the math operation word(s) in the problem window to the right, in either auto or drag & drop mode.*

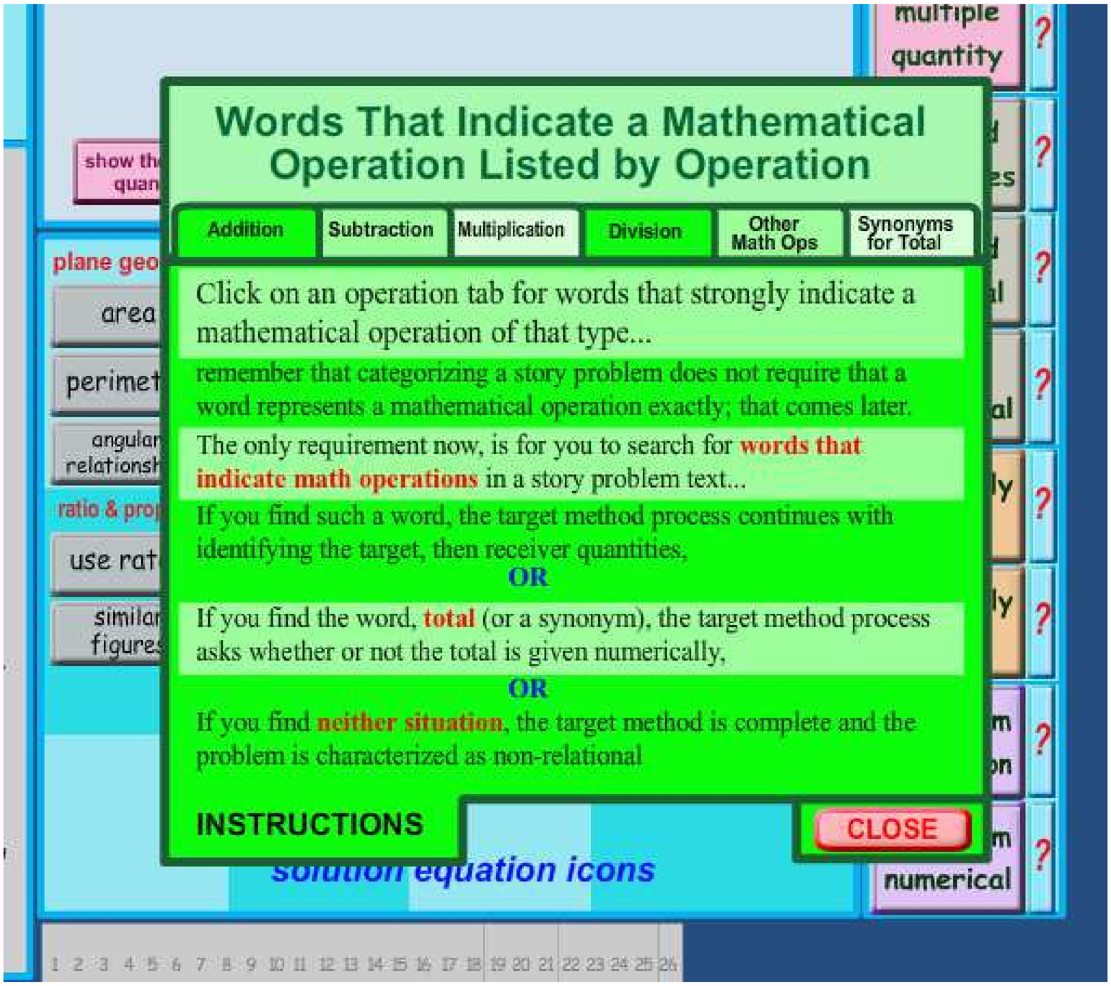
Currently Working This Problem...
Kendrick raises pigeons. He has six more than twice as many pigeons now than he had 6 months ago. If he has Y pigeons now, write an expression that represents how many pigeons he had 6 months ago.

Figure 39. The second characterization instructions appear in the instructions window, waiting for the user to read them, and then to click the NEXT button, continuing the worked example illustration. The steps in the categorization process proceed in this manner until all characterizations have been completed. At any time during the Auto Movement is ON mode, the user may click the BACK button to go back one step continuing to use either button for the desired result.

The second mode of the WEU is Drag and Drop is On. This mode duplicates the usage of MS-CPU in its manipulation of the characteristic icons. However, immediate feedback is given upon dropping an icon on a target which is felt to provide a better learning experience for a preparatory activity. The **BACK** button is disabled in this mode. Ideally, the student will begin using the WEU in the Auto Movement is On mode, switch back and forth between it and the Drag and Drop is On mode with advanced understanding, and use the Drag and

Drop is On mode exclusively for the last two or three problems (there are seven). The student then closes the WEU, returning to the MS-CPU possessing a familiarity with and knowledge of the categorization process.

19) Words that indicate mathematical operations (+, -, $\cdot$, ÷, ^{2}, ^{3}, √) are found in an always available wordlist as shown in Figure 40.



Words That Indicate a Mathematical Operation Listed by Operation

Click on an operation tab for words that strongly indicate a mathematical operation of that type...

remember that categorizing a story problem does not require that a word represents a mathematical operation exactly; that comes later.

The only requirement now, is for you to search for **words that indicate math operations** in a story problem text...

If you find such a word, the target method process continues with identifying the target, then receiver quantities,

OR

If you find the word, **total** (or a synonym), the target method process asks whether or not the total is given numerically,

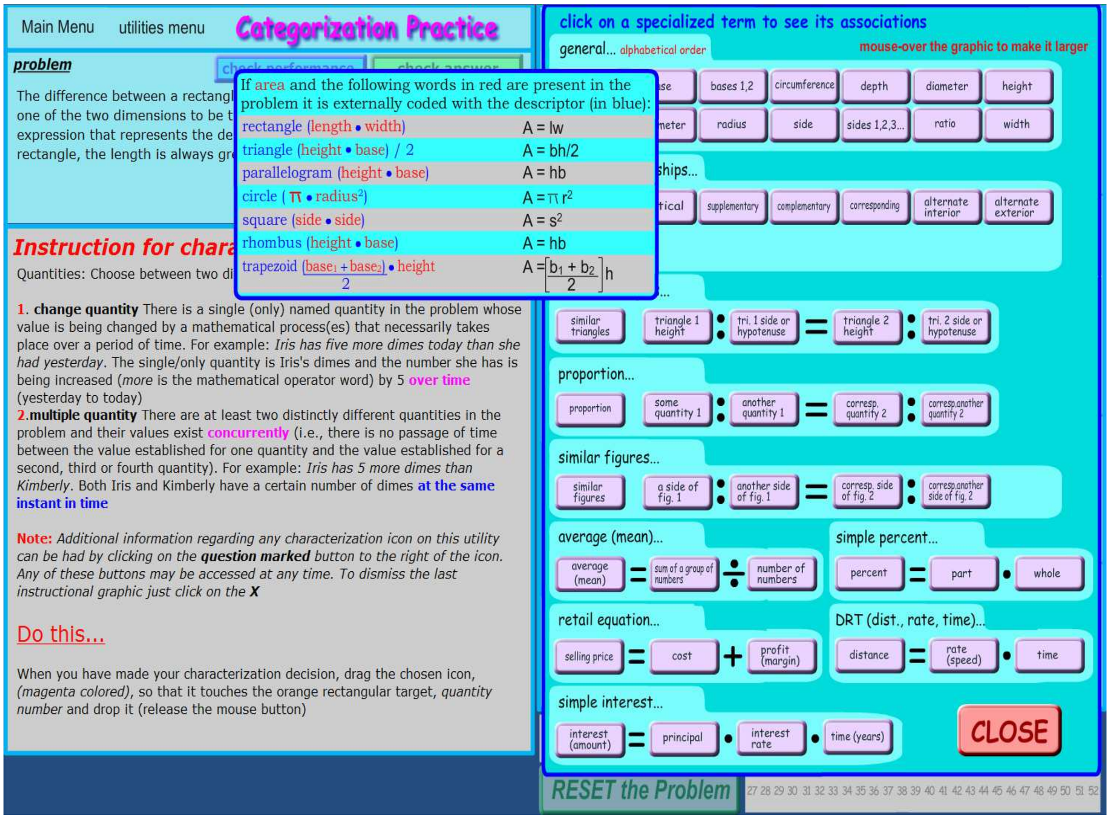
OR

If you find **neither situation**, the target method is complete and the problem is characterized as non-relational

Figure 40. As a sub-utility, a list of words in tab form is given, cataloged by the mathematics operation it is most closely associated with. Care is taken to negate the implication that these words absolutely mean the operation that they are listed under – only that they indicate some mathematics operation. Its absolute meaning in the context of the problem is determined by other means later in the problem-solving process. A wordlist is a valuable tool

in the Target Method of determining whether a story problem is *relational* or not by the presence or absence of mathematical operation words (MOP) in the problem text.

20) Specialized terms are words that name a unique quantity; one that can be found in a pre-existing solution equation (e.g., length, width, rate, time, profit, etc.). A directory of such terms and their association with other such terms is found on the *utilities menu* and is shown in Figure 41.



problem

The difference between a rectangl
one of the two dimensions to be t
expression that represents the de
rectangle, the length is always gr

If area and the following words in red are present in the problem it is externally coded with the descriptor (in blue):

rectangle (length • width)	A = lw
triangle (height • base) / 2	A = bh/2
parallelogram (height • base)	A = hb
circle (π • radius²)	$A = \pi r^2$
square (side • side)	$A = s^2$
rhombus (height • base)	A = hb
trapezoid $\frac{(\text{base}_1 + \text{base}_2)}{2}$ • height	$A = \left[\frac{b_1 + b_2}{2}\right] h$

Instruction for chara

Quantities: Choose between two di

1. **change quantity** There is a single (only) named quantity in the problem whose value is being changed by a mathematical process(es) that necessarily takes place over a period of time. For example: *Iris has five more dimes today than she had yesterday*. The single/only quantity is Iris's dimes and the number she has is being increased (*more* is the mathematical operator word) by 5 **over time** (yesterday to today)

2. **multiple quantity** There are at least two distinctly different quantities in the problem and their values exist **concurrently** (i.e., there is no passage of time between the value established for one quantity and the value established for a second, third or fourth quantity). For example: *Iris has 5 more dimes than Kimberly*. Both Iris and Kimberly have a certain number of dimes **at the same instant in time**

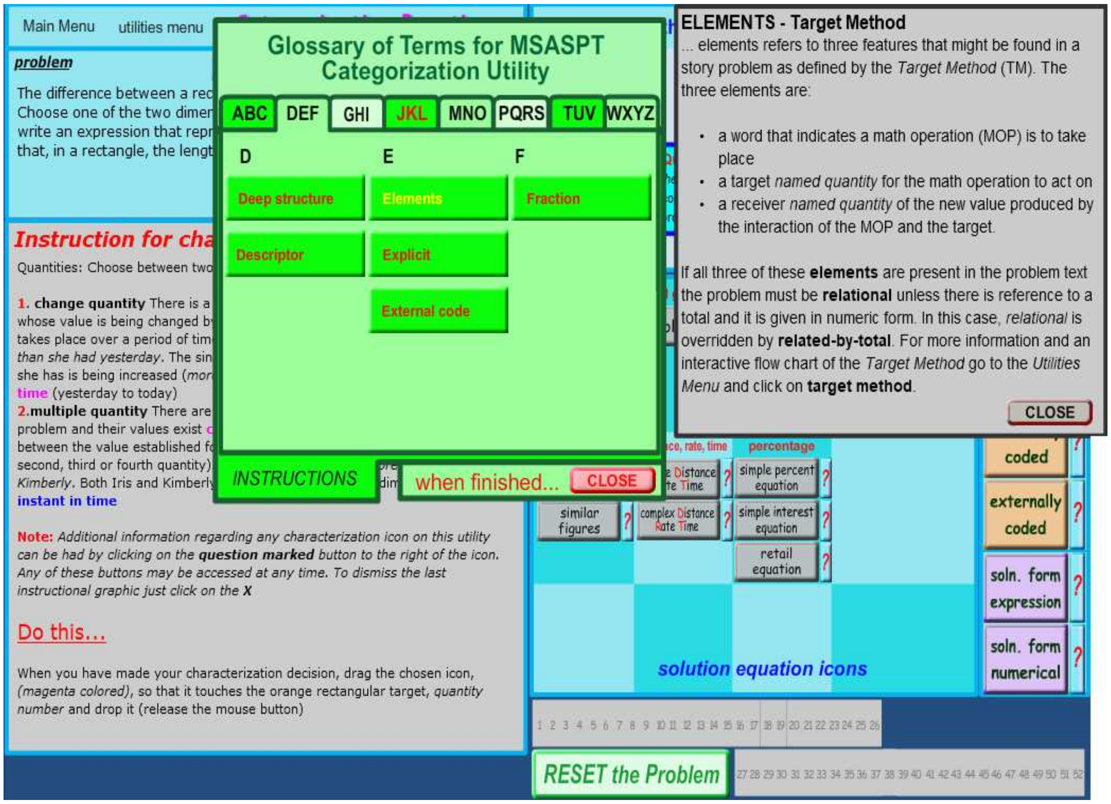
Note: *Additional information regarding any characterization icon on this utility can be had by clicking on the* ***question marked*** *button to the right of the icon. Any of these buttons may be accessed at any time. To dismiss the last instructional graphic just click on the* ***X***

Do this...

When you have made your characterization decision, drag the chosen icon, *(magenta colored)*, so that it touches the orange rectangular target, *quantity number* and drop it (release the mouse button)

Figure 41. A determination for the third characterization (solution equation source) as to whether a story problem is externally coded (a pre-existing solution equation) or not is a result of the examination of the quantity names present in the problem text. In this graphic, a specialized term, length, is present in the current problem. The specialized term utility is accessed from the utilities menu, and the term length is clicked. A graphic that names the quantities associated with length along with a descriptor (e.g., rectangle, triangle, circle, etc.) shows all possible solution equations in the GED Mathematics domain enabling the problem-solver to determine if all quantities and the descriptor are found in the story problem. The graphic has a mouse-over feature that enlarges it and is shown above in that state.

21) A compendium of all terms and phrases with special or unique definitions in the context of the MS-CPU are contained in a Glossary as shown in Figure 42.



problem

The difference between a rec
Choose one of the two dimen
write an expression that repr
that, in a rectangle, the lengt

Instruction for cha

Quantities: Choose between two

1. **change quantity** There is a
whose value is being changed b
takes place over a period of tim
than she had yesterday. The sin
she has is being increased (*mor*
time (yesterday to today)
2.**multiple quantity** There are
problem and their values exist c
between the value established fo
second, third or fourth quantity)
Kimberly. Both Iris and Kimberly
instant in time

Note: *Additional information regarding any characterization icon on this utility can be had by clicking on the* ***question marked*** *button to the right of the icon. Any of these buttons may be accessed at any time. To dismiss the last instructional graphic just click on the* ***X***

Do this...

When you have made your characterization decision, drag the chosen icon, *(magenta colored)*, so that it touches the orange rectangular target, *quantity number* and drop it (release the mouse button)

ELEMENTS - Target Method

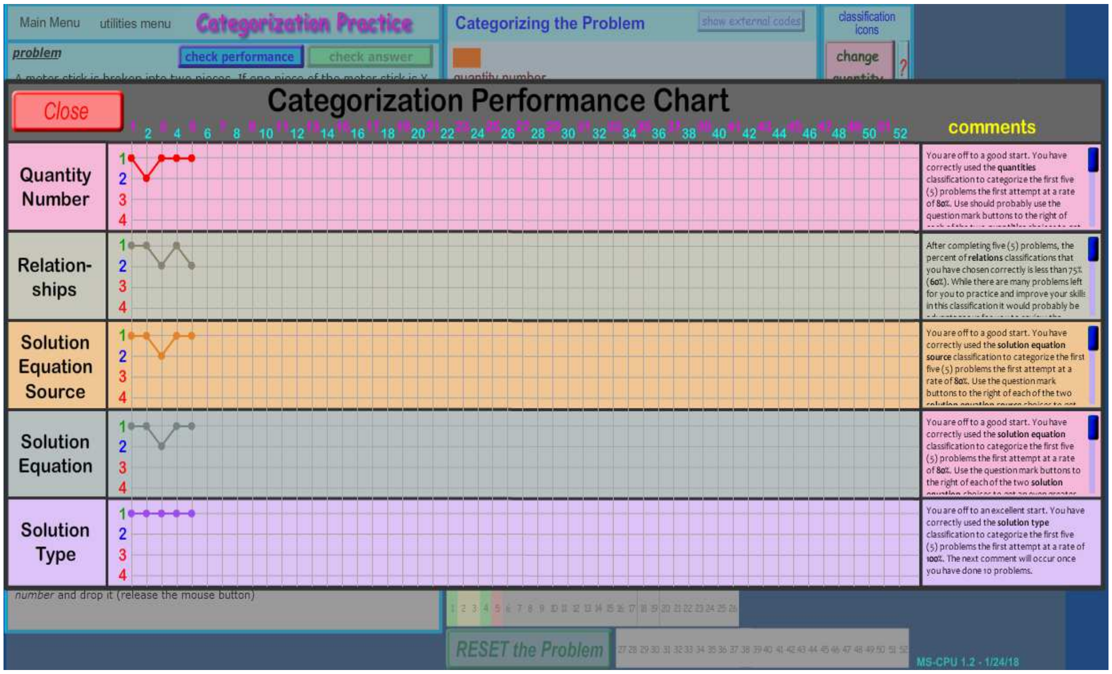
... elements refers to three features that might be found in a story problem as defined by the *Target Method* (TM). The three elements are:

- a word that indicates a math operation (MOP) is to take place
- a target *named quantity* for the math operation to act on
- a receiver *named quantity* of the new value produced by the interaction of the MOP and the target.

If all three of these **elements** are present in the problem text the problem must be **relational** unless there is reference to a total and it is given in numeric form. In this case, *relational* is overridden by **related-by-total**. For more information and an interactive flow chart of the *Target Method* go to the *Utilities Menu* and click on **target method**.

Figure 42. The figure shows the definition of *Elements* under the DEF tab.

22) A Student Performance Report Chart is offered at certain problem intervals as shown in Figure 43.



You are off to a good start. You have correctly used the **quantities** classification to categorize the first five (5) problems the first attempt at a rate of **80%**. Use should probably use the question mark buttons to the right of

After completing five (5) problems, the percent of **relations** classifications that you have chosen correctly is less than 75% (**60%**). While there are many problems left for you to practice and improve your skills in this classification it would probably be

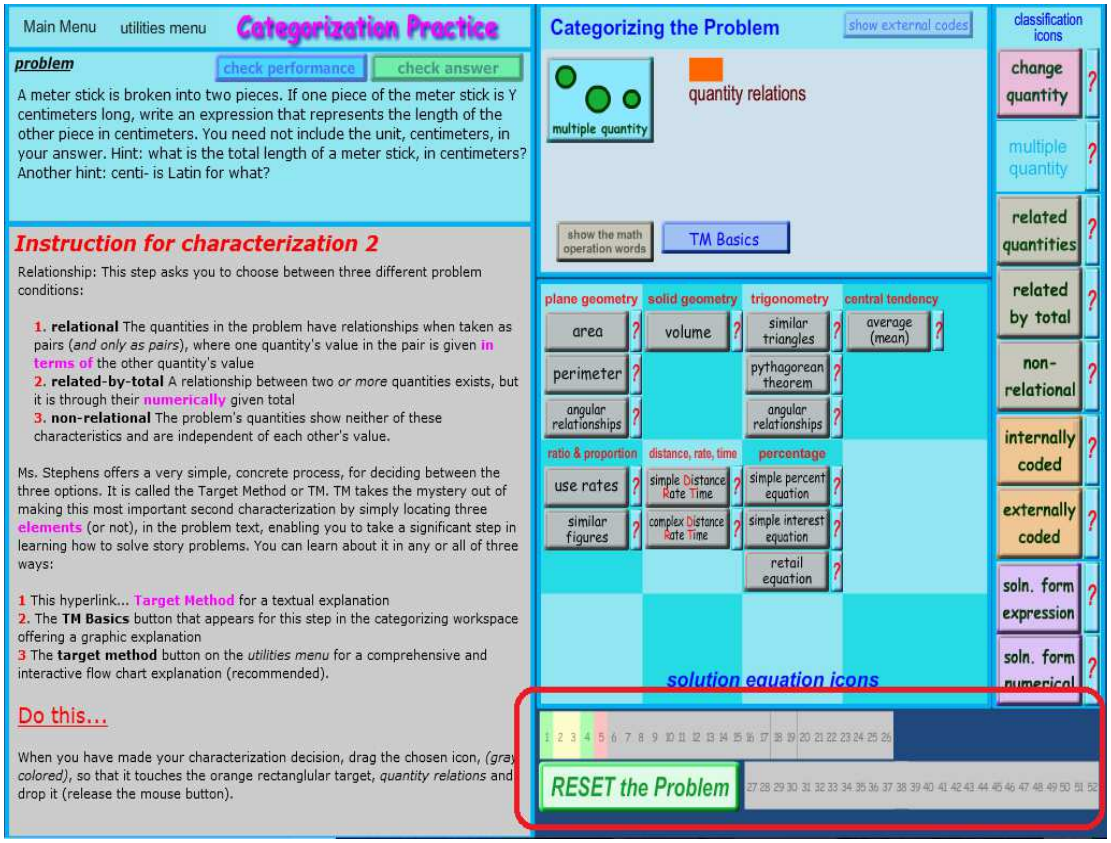
You are off to a good start. You have correctly used the **solution equation source** classification to categorize the first five (5) problems the first attempt at a rate of **80%**. Use the question mark buttons to the right of each of the two

You are off to a good start. You have correctly used the **solution equation** classification to categorize the first five (5) problems the first attempt at a rate of **80%**. Use the question mark buttons to the right of each of the two **solution**

You are off to an excellent start. You have correctly used the **solution type** classification to categorize the first five (5) problems the first attempt at a rate of **100%**. The next comment will occur once you have done 10 problems.

Figure 43. At problem intervals of 5, 10, 25, 40 and 52 – data is imported from the student model and represents knowledge component acquisition in the form of step performance. In this example, the student has completed the categorization of five story problems. The performance for each characterization appears in a graph with a comment on the performance from the tutor to the right of each graph.

23) A Progress Monitor displays categorization efficiency at a glance as shown in Figure 44.



problem

A meter stick is broken into two pieces. If one piece of the meter stick is Y centimeters long, write an expression that represents the length of the other piece in centimeters. You need not include the unit, centimeters, in your answer. Hint: what is the total length of a meter stick, in centimeters? Another hint: centi- is Latin for what?

Instruction for characterization 2

Relationship: This step asks you to choose between three different problem conditions:

1. **relational** The quantities in the problem have relationships when taken as pairs (*and only as pairs*), where one quantity's value in the pair is given **in terms of** the other quantity's value
2. **related-by-total** A relationship between two *or more* quantities exists, but it is through their **numerically** given total
3. **non-relational** The problem's quantities show neither of these characteristics and are independent of each other's value.

Ms. Stephens offers a very simple, concrete process, for deciding between the three options. It is called the Target Method or TM. TM takes the mystery out of making this most important second characterization by simply locating three **elements** (or not), in the problem text, enabling you to take a significant step in learning how to solve story problems. You can learn about it in any or all of three ways:

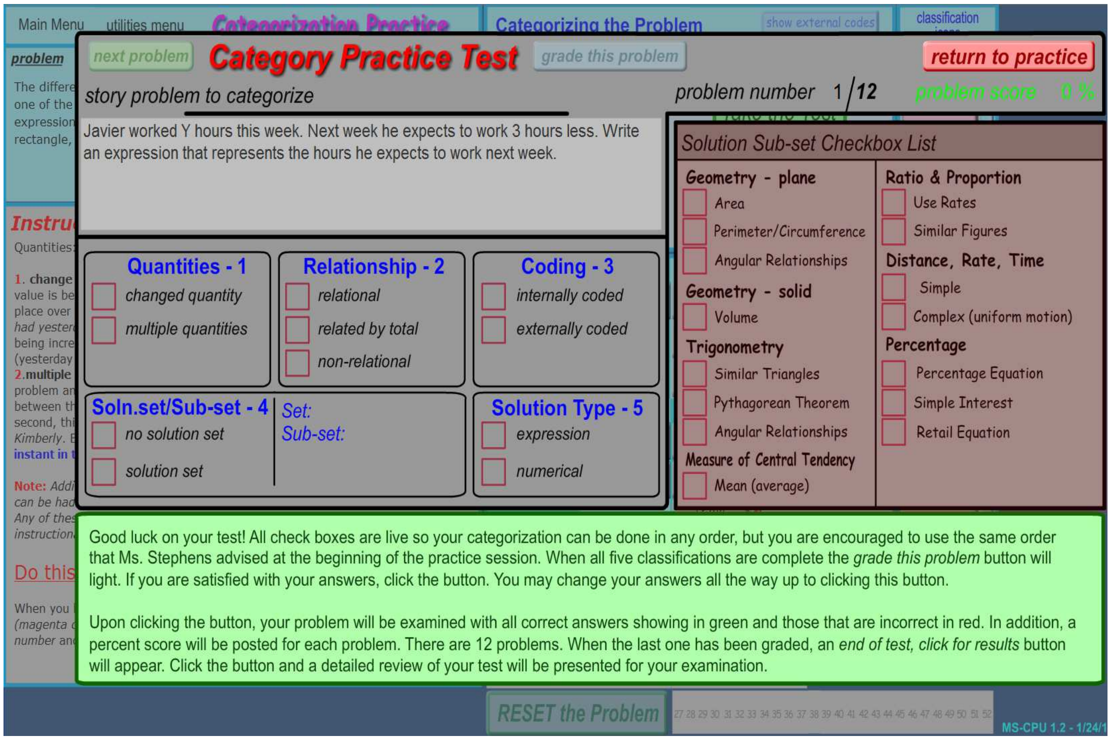
1 This hyperlink... **Target Method** for a textual explanation
2. The **TM Basics** button that appears for this step in the categorizing workspace offering a graphic explanation
3 The **target method** button on the *utilities menu* for a comprehensive and interactive flow chart explanation (recommended).

Do this...

When you have made your characterization decision, drag the chosen icon, *(gray colored)*, so that it touches the orange rectanglular target, *quantity relations* and drop it (release the mouse button).

Figure 44. The bands just below the WS-CPU indicate by color (green, yellow, red), performance on each problem as well as when a student is ready for testing. The example shows two problems done with no errors (green), two problems done with at least one error on the first try, but no errors on the second try (yellow), and one problem done with at least one error in each of the first two tries (red).

24) Once a sustained level of categorization proficiency is reached the student may elect to take a categorization test of twelve story problems as shown in Figure 45.



Javier worked Y hours this week. Next week he expects to work 3 hours less. Write an expression that represents the hours he expects to work next week.

Good luck on your test! All check boxes are live so your categorization can be done in any order, but you are encouraged to use the same order that Ms. Stephens advised at the beginning of the practice session. When all five classifications are complete the *grade this problem* button will light. If you are satisfied with your answers, click the button. You may change your answers all the way up to clicking this button.

Upon clicking the button, your problem will be examined with all correct answers showing in green and those that are incorrect in red. In addition, a percent score will be posted for each problem. There are 12 problems. When the last one has been graded, an *end of test, click for results* button will appear. Click the button and a detailed review of your test will be presented for your examination.

Figure 45. The student can elect to take the test when advised by the tutor (after ten consecutive correct categorizations) or after having completed all 52 problems. The test must be taken and passed with a score of at least 90%. If the test is not passed, the student model is consulted along with the test results to construct a remedial plan for the student. After completion of the plan, the student repeats a similar test of twelve problems. The checkboxes used on this test closely approximate the characterization selection mechanism used for the computer-administered GED Mathematics test as well as duplicating the categorization UI on the MSASPT.

This completes Section VII and the Introduction. The following Methods chapter will describe how the MSASPT Categorization Step will use formative research to produce an optimum design for teaching GED students how to categorize story problems for use in constructing and retrieving schemas containing the declarative and procedural knowledge required to solve a certain type of problem.

CHAPTER 2

RESEARCH METHODS

Section 1: Purpose of the Study:

The purpose of this study was to identify, from the perspective of the GED Test affiliated study participant, the particular modifications to the Ms. Stephens' Categorization Practice Utility (MS-CPU) learning environment that would provide the greatest possible effectiveness, efficiency, and appeal for the participant. These modifications have prepared the system for further study as to its efficacy in teaching GED Mathematics Test Preparation students a story problem categorization taxonomy using a yet to be determined method of summative research. Such modifications are broadly made according to their user interface or pedagogical association.

Section 2: Research Approach:

The MS-CPU facilitates the categorization of story problems using a taxonomy that serves both as a problem type identifier and an initial appraisal of its deep (solution) structure. The utility teaches the student how to take the critical first step in a larger tutoring system used to teach story problem solution over five steps. Both of these tutoring systems are innovative in their pedagogical principles as they work to circumvent the previously discussed inability of many students to assimilate conceptually centered story problem solution teaching. This unique pedagogy requires a very strong synergistic and intuitive setting for the many components of the user interface.

The extensive variability in this system involving a unique pedagogy, critical help timing, availability of multiple help streams, idiosyncratic terminology, and intuitive use, required an initial level of research to prepare the system for use in subsequent studies that would more quantitatively address its effectiveness. The framework chosen for conducting this initial research was the project implementation category of formative research. By definition, formative research (or evaluation) takes place before or during a project's implementation with the goal of improving a project's design and performance.

Using largely qualitative methods that are verbally and visually sourced, descriptive information was collected and used to apprehend the aspects of MS-CPU program that caused extraneous cognitive load (ECL). Additional information collected in this manner as well as designer input was used to ascertain why an aspect worked or why it didn't, and then used to mitigate the identified difficulty through modification of the system's user interface, re-alignment with or adjustment to pedagogical principles, or both.

The descriptive information was produced using a collaborative approach between the participant and the investigator with questions and answers originating from both as the participant used the MS-CPU. Notes were taken by the investigator under various headings for the categorization practice utility and its sub-utilities, producing a rich collection of needed change descriptions and a keyword or two for its possible mitigation as well as any positive aspects of the program the participant felt strongly enough to note. A very productive source of needed change was the in-depth interview (IDI) conducted after the post-test. The questioning regarding each of the five characterizations about how they arrived at their answers yielded the underlying pedagogical causes for the participant's characterizing misidentifications.

Section 3: Participants:

Study participants were purposefully drawn from the general adult population with the stipulation that each had been involved in a GED Test Preparation regimen. There were no other conditions attached to study participation. The research location requirement was simply that the site was in reasonable proximity to the participant's residence and possessed a physical environment that provided for privacy and concentration on the task. All but one research session took place at libraries in the Phoenix area that offered reservations for at least two hours of time in a study room in the physical building. The session that did not take place in a library study room was conducted in the participant's home due to parental responsibilities. There were two iterations of the study, each requiring three participants for a total of six that met the selection stipulation of having been involved in a GED Test Preparation regimen.

Each participant was observed using the MS-CPU. The observation session was the central method of the study, was collaboratively open-ended and lasted for 70 to 80 minutes. The participant was encouraged to think aloud and ask questions even as the investigator asks questions and explains both pedagogical and user interface difficulties when the participant shows signs of confusion. The collaborative nature of the session led in many different directions through each study iteration, providing critically rich information with which to engineer the system for greater effectiveness and appeal.

Section 4: Data Collection Tools:

Data collection instruments consisted of an open-ended investigator observation form (Appendix A) for each of the following major utilities: the Main Categorization Utility; the Worked Examples utility; the Target Method Flowchart; the Solution Equation (Code) Source Flowchart. Each observation form was divided into five sections: Comprehension of Instructions; Comprehension/Use of Definitions; Categorization Flow; User Interface Intuitiveness; Miscellaneous Observations. A 12 story problem pre-test (Appendix B) and a 12 story problem post-test (Appendix C) were used to measure improvement in the participant's ability to categorize story problems using the given taxonomy and also to reveal deficits in the system pedagogy. Problems were worded differently but categorized similarly across both tests. Finally, a Likert scaled survey (Appendix D) for each of the four major utilities named above was constructed with a varying number of questions dependent on the utility or flowchart.

Section 5: Procedures:

Final Arizona State University IRB approval was granted on the 2nd of August, 2017 (Appendix S) and research began on the 11th of September with the first participant. Two iterations of the study were conducted with each research session following the same event sequence once the preliminary study information sheet was read by the participant and the consent form signed. After each iteration, the data collected was used to mitigate system difficulty before the next iteration was conducted. The second mitigation was expected to prepare the system for more directed research. The session began by asking the study participant to read a short informational page regarding the purpose and nature of the study (Appendix E). A description of the story problem categorization protocol was read (5-8 minutes) describing how each of the five characterization decisions that make up the taxonomy is arrived at (Appendix F) followed by a paper and pencil pre-test of 12 story problems for the participant to categorize.

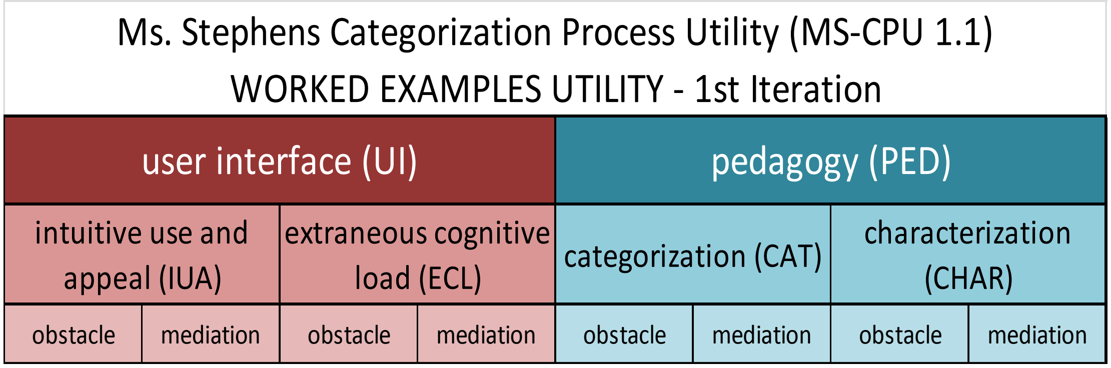
A collaborative observation period of 70 to 80 minutes began following completion of the pre-test with the participant encouraged to think aloud and ask any and all questions that came to mind regardless of participant-determined import. The investigator, as well, asked questions and gave operational advice as deemed appropriate in a conversational atmosphere conducted to discover the difficulties present in the system from the participant's perspective. Data were recorded in the form of notes written in the relevant section of the Observation Guidelines (Appendix A) and collected continuously throughout the observation period and for 5 – 10 minutes after the observation session to annotate previous entries for clarity and context.

Additional data was collected from a survey (Appendix D) for the major utilities named above in the Data Collection Tools section (Main Categorization Utility; the Worked Examples utility; the Target Method Flowchart; the Solution Equation (Code) Source Flowchart). The post-test was another source of data as incorrect answers in concert with an additional conversation with the participant exposed learning dissonances that are present in the pedagogy and need remediation. Other, quantitative data were collected from the difference in the pre-test (Appendix B) and post-test (Appendix C) scores for each characterization and for each iteration and for post-test scores from each iteration.

Section 6: Data Analyses:

Data was collected identically for each of two iterations from observation sessions, surveys, and informal post-hoc interviews and consisted of the event that caused an extraneous cognitive load or some other difficulty and its location within the system. The originator of the event may have been the participant or the investigator and may have been the product of observation, a question, or an answer. The analysis proceeds from eight tables: four for the first study iteration concerning the four major utilities and another four for the second study iteration concerning the same four major utilities. Each table had three qualifying rows. The top header was either user interface (UI) or pedagogy (PED). The second-row sub headers were intuitive use - appeal (IUA) and extraneous cognitive load (ECL) under the UI header. Under the PED header were the categorization (CAT) and characterization (CHAR) subheaders. CAT referred to the overall categorization process while CHAR referred to each of the five characterizations that constitute a story problem categorization.

The third row of headers was a pairing of table cells that contain the problematic data on the left (the obstacle) and the method of correction (the mediation) on the right. Each of these four pairs lies in continuous columns below the second-row subheader (IUA, ECL, CAT, and CHAR) as shown for the Worked Examples first iteration by the abbreviated Table 1 below. The eight tables, each headed as in Table 1 are located in the Appendices G – N.



Ms. Stephens Categorization Process Utility (MS-CPU 1.1)
WORKED EXAMPLES UTILITY - 1st Iteration

user interface (UI)				pedagogy (PED)			
intuitive use and appeal (IUA)		extraneous cognitive load (ECL)		categorization (CAT)		characterization (CHAR)	
obstacle	mediation	obstacle	mediation	obstacle	mediation	obstacle	mediation

Table 1.

To measure the overall effectiveness of the obstacle mitigation implemented on the system when taken as a single entity, pre-test scores for the first iteration's participants were averaged along with post-test scores. A comparison of the two average scores was used to determine whether participant performance improved from pre- to post-test for the first iteration. The same procedure was performed for the second iteration. A second comparison between participant performance improvement scores for each iteration was performed to determine whether improvement in pre- post-test scores between iterations occurred. See Table 2, pg. 103, in the Findings chapter.

To measure the effectiveness of the obstacle mitigation implemented on the system as it relates to a specific characterization, participant pre and post-test average score differences for each characterization were measured in the first iteration and participant pre and post-test average score differences were measured likewise in the second iteration. The average differences for each characterization for each iteration were then compared to one another. See Table 3, pg. 107, in the Findings chapter.

Research session-ending Likert-scaled surveys concerned with each of the four major utilities mentioned above and for each iteration were administered to provide additional depth to the observational data. The participant replies to a total of 44 questions were entered into an Excel table structured to produce the percentage for each of the six response types (not used, strongly disagree, disagree, no opinion, agree, and strongly agree) along the Likert scale. The analysis was done to compare the differences between corresponding percentages associated with each iteration. See Table 4, pg. 108, in the Findings chapter.

Section 7: Ethical Considerations:

Except for the signed consent document, no study instrument was directly or indirectly associated with the identification of the participant. All completed study materials were kept by the investigator in a secure environment. No previously collected study information was disseminated in any form to any subsequent participant.

Section 8: Trustworthiness:

Observation and conversation throughout the study provided ample evidence that the experiential similarity of the study participants to the GED Test population target demographic produced a commitment to an honest performance from the study participants. Additionally, shared economic and social difficulties and disadvantages tended to strengthen this commitment and produce a study, dependent on an honest evaluation of the MS-CPU, which was valid and trustworthy.

Section 9: Potential Research Bias:

The potential for investigator bias did exist as the object of the study, the MS-CPU, was designed and developed by the investigator. However, the motivation to produce an effective system for teaching story problem solution came from a heartfelt desire to increase the number of GED graduations. Any possessed bias as to the dismissal, or deferential treatment of the data was fully managed by the investigator's desire to provide the best possible opportunity to pass the GED Mathematics Test for GED Test Takers.

Section 10: Limitations:

Time constraints for each session (2 - 2½ hours) prevented each participant from using all of the various utilities and sub-utilities in the system. Care was taken to make certain that all of these experiential programs were used by at least one participant over the two iterations of the study.

CHAPTER 3

FINDINGS

Section 1: General:

The MS-CPU primary user-interface contains a story problem window, an instructional window, and a workspace (Figure 5, pg. 44). Within this user-interface, the Main Menu provides access to the operational features of the system (Figure 19, pg. 59) and the utilities menu provides access to major utilities available for additional explanation and practice (Figure 20, pg. 60). Just-in-time clickables (buttons – lower left of Work Space and hyperlinks in the instructional text) placed proximally to the task serve to access several other sub-utilities in addition to definitions for phrases and terms that have specific meaning in the MS-CPU categorization process (Figure 29, pg. 68). All of these utilities and sub-utilities serve to concretely illustrate and define the decision-making process for each of the five characterizations that make up a story problem's category and its solution appraisal as well as serving to lower intrinsic cognitive load with a modular rather than molar pedagogical approach (Gerjets, et al., 2004).

The goal of this formative research was to maximize the MS-CPU tutoring system's effectiveness, efficiency, and appeal as experienced by people who would have had the occasion to use it in their preparation to take the GED Mathematics Test were it available. Each of the six participants was able to work through three and sometimes more of the story problems contained in either the main categorization practice utility or in the worked examples utility. The qualitative measures conducted over two iterations showed a decided improvement in the flow of meaningful information between the participant and the tutoring system and the accessibility of such information from the first iteration to the second. This improvement is supported by an approximately 80% reduction in the number of obstacles encountered by the participants, the increased depth of exploration into the tutoring system, and a diminished level of directional and instructional discourse between the participant and the investigator as observed by the investigator in the second iteration by each participant.

An analysis of the quantitative data collected from a test given to each participant after reading a short, written description of the categorization taxonomy (the pre-test) but before using the MS-CPU and a test given after using the MS-CPU for approximately 70 minutes (the post-test) provided some corroborative evidence as to whether obstacles to understanding and use had been mitigated between iterations and whether characterization pedagogical difficulties persisted from iteration to iteration. These analyses can be found in the section titled Results from Pre- and Post-test Performance on pg. 102 in this chapter.

The appeal of the user interface to the participant remained consistently very favorable through the entire research period which began on the 11th of September and ended on the 13th of February. What few user-interface design weaknesses that occurred were easily and effectively mitigated. The overall appeal of the system, including its pedagogy, was typified by one participant's response near the end of the research session: "I wish I could spend more time with this. My daughter teaches GED [test preparation], and she would love it." An analysis of survey responses suggested the positive appeal as well.

Findings from Observations, Think-Alouds, Conversations:

The interpersonal communication session (observations, think-alouds, and conversations) is the major aspect of this MS-CPU formative research. The findings from these sessions are briefly enumerated in the following paragraphs according to their association with the Main Categorization Utility or one of the three supporting utilities and labeled according to the table headings described in the Methods Section for each iteration in the sequence. The qualifying frequency of a particular obstacle is once. For a detailed account of all obstacles and their mitigations for each utility see Appendices G - J for the first iteration. For the second iteration see Appendices K - N. Pre and post-test, as well as survey results, are considered from the perspective of iterations in the following sections. The findings in this section do not consider these perspectives.

Section 2: Main Categorization Utility (MS-CPU • Figure 1 – pg. 43)

User Interface; Intuitive Use and Appeal Findings:

The ability to know the correct physical response or make the correct choice as each step in a process is encountered without significant consideration is called intuition or making an intuitive decision. However, people possess varying degrees of intuition, largely dependent on experience and not always correct. A concept that proceeds from an intuitive choice is making an obvious (Marshak, 2010) or even only, choice. Obvious, in this context, virtually ensures that a correct and timely choice will be made. In designing and developing an intelligent tutoring system, creating an obvious or only choice instead of intuitive choice may require significantly more engineering and thus more development time. In many cases, the obvious over intuitive design feature is well worth the additional time and programming.

For example, in the first research version of MS-CPU (1.1), it was suggested, then logically and historically supported in the initial general instructions, that the participant use the Worked Examples utility before using the MCU. The suggested path was not taken by any of the three study participants. For the second iteration of the study, using MS-CPU 1.2, the general categorization instructions are presented immediately (not menu driven) for a certain pre-determined period, after which, the Worked Examples utility is automatically presented. The participant may not exit this utility until at least one problem is done, effectively presenting an obvious/only choice as to whether the Worked Examples utility is accessed.

Other methods for obvious choice used in version MS-CPU (1.1): animated arrows pointing to a button that should be selected; grayed-out/disabled buttons with mouse-over messages advising the user to make a different selection; a fuchsia colored practice hyperlink in the instructions explained in a separate paragraph; and buttons placed proximally to the object of an action.

In systems that have a significant degree of complexity, such as MS-CPU, it is virtually impossible to make all choices obvious. The remaining choices in this system have been engineered to be intuitive, but the observation and conversations that took place in this phase of the research demonstrated that the ability to make intuitive decisions is relative to experience. Experience in categorizing story problems or even using a computer is in short supply in the GED student population. As related later in the Discussion Section, a significant training period on the use of the system would be highly recommended. Certainly, the appeal of any teaching system is enhanced by the free flow of information between the tutoring system and the student is facilitated by knowledge of how to use it.

User Interface**;** Extraneous Cognitive Load;

Extraneous cognitive load (ECL) is defined as the cognitive load generated by the manner in which information is presented to learners (Chandler & Sweller, 1991). Instances in the MS-CPU 1.1 tutor that created ECL by the way in which they were presented fall into four categories. (1) Use of terms and phrases that are not in common usage (2) Graphical representations that are incomplete or are capable of multiple interpretations (3) Non-availability of needed utility or sub-utility designed to concretize pedagogical concepts (4) User is unaware that a utility or sub-utility is available to concretize pedagogical concepts.

An example of the first category is a word or phrase whose contextual definition is unclear or non-existent and must be hyperlinked to its definition as well as added to a Glossary available at all times for such a word or phrase. A second category example is the graphical manipulative for the named quantities number/*multiple quantities* characterization having three different sized circles to represent more than one quantity. Two of the participants interpreted the circles to mean that there were three named quantities in the problem. Looking for three quantities when there were two or four led to confusion in identifying each quantity and misapprehension of the principles behind the characterization.

A third category example is a difficulty in identifying whether or not a mathematical operation word or phrase was present in the problem text. While the phrase *mathematical operation word* is hyperlinked to its definition, the actual word or phrase needed to be highlighted as such in the problem text to mitigate the ECL. Further examination of this particular mitigation can be found in the Discussion chapter (pg. 124 – final paragraph) as there is some pedagogical risk associated with this system modification. A fourth category example is a subject that is not aware of a just-in-time button available in the workspace that graphically demonstrates how to make a second characterization decision.

Pedagogy; Categorization;

There were no issues in the pedagogical principles put forth by the participants in either the instructions to begin using the MS-CPU. Neither were there any concerns with the rationale for using the particular categorization taxonomy presented in the MS-CPU or the order in which the story problem characterizations that constitute the taxonomy were presented.

Pedagogy; Characterization;

As expected, all five characterizations (quantity number, quantity relationships, solution equation source, solution equation, and solution type) initially presented difficulties to the participants. For example, a pedagogical obstacle occurred in the solution equation source characterization. The criteria for the decision (internal or external) needed to include problem information that is presented in the form of a proportion as a criterion for an externally sourced equation (coded), even though the named quantities are not specialized quantities (e.g., area, length, height, speed, etc.). See Appendices G (1st iteration) and K (2nd iteration) for more detailed MCU obstacle and mitigation information.

Section 3: Target Method Flowchart Utility (TM • Figure 14 – pg. 59)

User Interface; Intuitive Use and Appeal Findings;

The Target Method (TM) is designed to algorithmically facilitate an accurate second or *quantity relations* characterization. The TM utility is an interactive flow-chart that presents the Target Method as a series of decisions. These decisions are used by the student to expose whether a story problem contains at least one pair of named quantities where one quantity's value is given in terms of the other named quantity (*relational*), or the total of two or more named quantities is given in numerical form (*related-by-total*), or neither condition exists (non-relational).

While using the TM utility, participants encountered three counter-intuitive obstacles as detailed in Appendices H. An example: the path taken based on each decision choice is displayed in the form of a black arrow leading from the made decision to the next decision. When the user made a correct decision, the arrow turned blue and connected the two decisions; when an incorrect decision was made the arrow remained black. To enhance the utility's intuitive use, the arrows were color-coded green for correct and red for incorrect, allowing the correctly completed path to stand out vividly in green and providing to the user, intuitively, the goal of the flow-chart.
User interface; extraneous cognitive load (ECL) (TM);

In the early 1970's Noel Burch created his stages of competency working with Gordon Training International ("It's Time to Give Noel Burch", n.d.). The fourth and final stage, unconscious competence, occurs when a skill, concept, process, etc. is learned so well that the learner does not have to think about its execution or meaning. Instructional designers must be very careful to make certain that their unconscious competency does not result in an aspect of a user interface or pedagogy that assumes a level of knowledge that does not exist for the user. For example, the term used to describe the TM Utility, "flowchart", is not a broadly used term and presented an ECL obstacle for all three of the first iteration users. Mitigation was accomplished by defining the term "flowchart" at the very start of the utility using a text window that automatically appeared and persisted for 45 seconds while the user read an extended definition of the term.

Pedagogy; Categorization (TM)**;**

This utility is concerned with the second characterization only and has no direct connection with pedagogical considerations for the categorization taxonomy. See below for characterization findings in the TM Utility.

Pedagogy; Characterization (TM)**;**

The first step or decision to be made when using the Target Method (and the TM Utility) is to look for a word or phrase that connotes a mathematical operation (e.g., less, more, greater than, etc.) or the word total (or a synonym, e.g., sum, altogether, makes, etc.). If a mathematical operation word or phrase (MOP) was found, subsequent steps became problematic. The instructional text for identifying which quantity in the named quantity pair is the *target* of the MOP or its *receiver* was too conceptual. Incorporating the principles attendant to concrete articulation (numerical value given to an unknown value) the TM Utility design becomes much clearer as to how these opposing quantities are identified and the characterization of the problem as *relational* is verified. See Appendices H (1st iteration) and L (2nd iteration) for more detailed TM utility obstacle and mitigation information.

Section 4: Solution Equation Sources Flowchart Utility (SES • Figure 16 – pg. 61)

User Interface; Intuitive Use and Appeal Findings;

The Solution Equation Sources instruction is designed to algorithmically facilitate an accurate third or *solution equation source* characterization as to whether the solution equation for a particular problem pre-dates or pre-exists the problem at the time it is being read, and the problem-solver needs only to be aware of it or the solution equation must be synthesized on-the-spot and entirely and only from the information in the problem. The Solution Equation Sources Utility is an interactive flow-chart that presents the instruction as a series of decisions. These decisions are used to expose whether a story problem's named quantities are congruent with the named quantities in a pre-existing solution equation (e.g., a problem has area, length, width, and rectangle named quantities; the solution equation for the area of a rectangle – A = lw – matches exactly). Its design converts this match/no match process into an algorithm. If there is a match, the problem characterization is *externally coded*, if not the problem is *internally coded*.

In spite of every effort to make the choices to be made in a user interface intuitive and even obvious, if possible, user errors will occur. When they occur, the interface should respond to the error in some way that succinctly explains how the error occurred and how to operate the interface so that it does not occur again. For example: clicking YES or NO at each decisions point in the flowchart can only occur at certain times. Rather than simply not responding to the click and leaving the user unaware of the cause for the inactive button, a transparent user interface object covers the button, receives the click, and responds with an explanatory error message.

User interface; extraneous cognitive load (ECL) (SES);

Excessive amounts of text presented to the user seriously increase ECL and many, if not all, will skim or even refuse to read the information, leaving the user ignorant of important principles and directions. For example, the instructions for using the Solution Equation Sources Utility presented much more material than the user can or needs to process. The mitigation of this obstacle involved removing unnecessary detail and giving an overview of the process instead. Step-by-step instructions and relevant information were shown proximal to the required action to eliminate the split-attention effect.

Pedagogy; Categorization (SES);

This utility is concerned with the third characterization only and has no direct connection with pedagogical considerations for the categorization taxonomy. See below for characterization findings in the Solution Equation Sources Utility.

Pedagogy; Characterization (SES);

The characterization of a story problem as *externally coded* using the criteria of a match between named quantities in a story problem and a pre-existing solution equation (or code) proved to be incomplete. Participant 1A attempted to make this characterization on a story problem containing four quantities not found in any pre-existing solution equation which would have resulted in an internally coded characterization. However, the quantities (parts of epoxy, parts of hardener) were given in the form of a proportion. A proportion is a pre-existing solution equation so the problem is *externally coded* and the additional criteria to mitigate this obstacle was written into the pedagogy. See Appendices I (1st iteration) and M (2nd iteration) for more detailed SES obstacle and mitigation information.

Section 5: Worked Examples Utility (WE • Figure 23 – pg. 68) Findings:

The WE utility was designed to behave, look, and impart the same knowledge that the MS-CPU does with the additional feature of a demonstration or automatic mode where the user can watch the characterizations being made by the computer at the click of a *NEXT* button (or *BACK*). The user can also transition gradually to an autonomous mode (drag and drop) in preparation for a return to the MCU. All of the obstacles present in the MCU were mitigated as well in the WE utility.

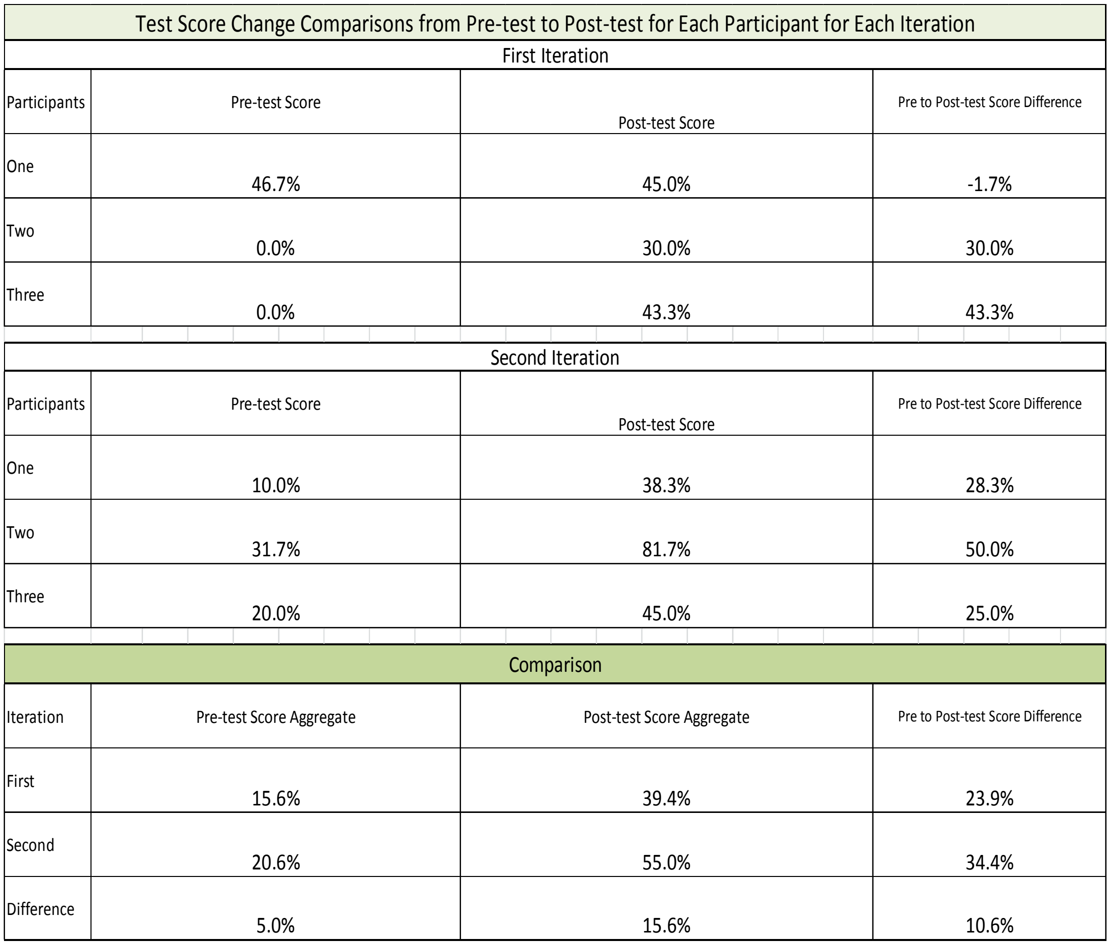
As a result of the additional modes of use, obstacles that weren't present in the MCU were present in the WE utility. For example, the NEXT and BACK buttons were permanently located along the upper margin of the utility, negatively impacting the user's intuitive sense as to where they might be. This obstacle was mitigated by buttons that adjusted their location to always be at the bottom of the instructions for a particular step in the categorization process. See Appendices J (1st iteration) and N (2nd iteration) for more detailed WE utility obstacle and mitigation information.

Section 6: Pre- and Post-Test Performance Results:

Participants (3 for each of the two iterations) took a 12 question pre-test after reading an explanation of the categorization taxonomy that included information about how to make each of the five characterization decisions based on certain criteria. After the pre-test and for a period of 70 – 80 minutes, each of the subjects used the MS-CPU with the investigator present as an observer and a participant in conversations about the tutoring system. At the conclusion of this session, the post-test was taken. A minimal score on the post-test was established at 65% from the notion that continued experience with a teacher qualified to use the tutor and the tutor itself, a score of 95-100% is attainable.

The aggregated pre- to post-test (P-PT) data for each iteration analysis was used to examine the effectiveness of the modifications made between the first and second iterations (See Table 2, below). For the first iteration, the average pre-test score was 15.6%, and the average post-test score was 39.4% for a P-PT score improvement of 23.9%. For the second iteration, the average pre-test score was 20.6%, and the average post-test score was 55.0% for a P-PT score improvement of 34.4%. The 10.6% improvement from the first iteration performance to the second suggests to a significant degree that the modifications made to the system based on the observational data collected after the first iteration produced a more effective tutor for use in the second iteration.

TABLE 2



Test Score Change Comparisons from Pre-test to Post-test for Each Participant for Each Iteration

First Iteration

Participants	Pre-test Score	Post-test Score	Pre to Post-test Score Difference
One	46.7%	45.0%	-1.7%
Two	0.0%	30.0%	30.0%
Three	0.0%	43.3%	43.3%

Second Iteration

Participants	Pre-test Score	Post-test Score	Pre to Post-test Score Difference
One	10.0%	38.3%	28.3%
Two	31.7%	81.7%	50.0%
Three	20.0%	45.0%	25.0%

Comparison

Iteration	Pre-test Score Aggregate	Post-test Score Aggregate	Pre to Post-test Score Difference
First	15.6%	39.4%	23.9%
Second	20.6%	55.0%	34.4%
Difference	5.0%	15.6%	10.6%

P-PT data was collected for each of the five characterizations over the two iterations as well. Analyses of the performance changes from P-PT and from iteration to iteration proved unreliable, with only three subjects in each iteration, for finding causal consideration for characterization improvement over the two tests and iterations. However, post-test results by themselves from iteration to iteration were powerful enough to further identify, in concert with the observational data, short-comings that continued to persist through the study in the user-interface and pedagogy of a particular aspect of the categorization taxonomy. A review of the results (See Table 3) for each characterization from the perspective of persistent difficulty for subjects through both iterations follows:

Quantities Number (change or multiple) – first characterization. The student must decide whether the problem has a single quantity given in two states, usually now and before or after, (change) or two or more discrete quantities that exist in the problem at the point in time – now (multiple). First iteration average post-test score (correct) for identification of *change quantity* story problems was 16.7% and the second iteration average score for the same characterization was 50.0%. The 33.3% difference from one iteration to the next suggests that the modifications made to the pedagogical presentation that supports the student perception of a *change quantity* story problem were reasonably effective but in need of further enhancement to reach the 65% threshold. *Multiple quantity* story problem identification average score for the first iteration was 56.7% and 80% for the second iteration for an improvement of 23.3%. These results suggest a pedagogical presentation that is effective, and with continued tutor experiences, would produce a very satisfactory student performance in identifying *multiple quantity* story problems.

Quantity Relations (relational, related-by-total, or non-relational) – second characterization. The student must decide whether the problem contains at least one pair of quantities where one quantity is given in terms of the other quantity (*relational*) or at least two quantities whose total is numerically given (*related-by-total*) or neither of these conditions are true (*non-relational*). First iteration average score for identification of a story problem as *relational* was 33.3%, and the second iteration average score for the same characterization was 60.0%. The 26.7% difference from one iteration to the next suggests that the modifications made to the pedagogical presentation that supports the student perception of a *relational* story problem were reasonably effective but in need of further enhancement to reach the 65% threshold.

The score for identification of a story problem as *related-by-total* for the first iteration was 44.4% and 55.6% for the second iteration for an improvement of 11.2%. These results suggest that the modifications made to the pedagogical presentation that supported the student perception of a *related-by-total* story problem were reasonably effective but in need of further enhancement to reach the 65% threshold. The average score for identification of a story problem as *non-*

relational was the same for both iterations at 25%. While this characterization showed no improvement over the two iterations, the decision to use this identification is based entirely on an absence of either of the first two conditions and as the first two condition's effectiveness improves so will this condition.

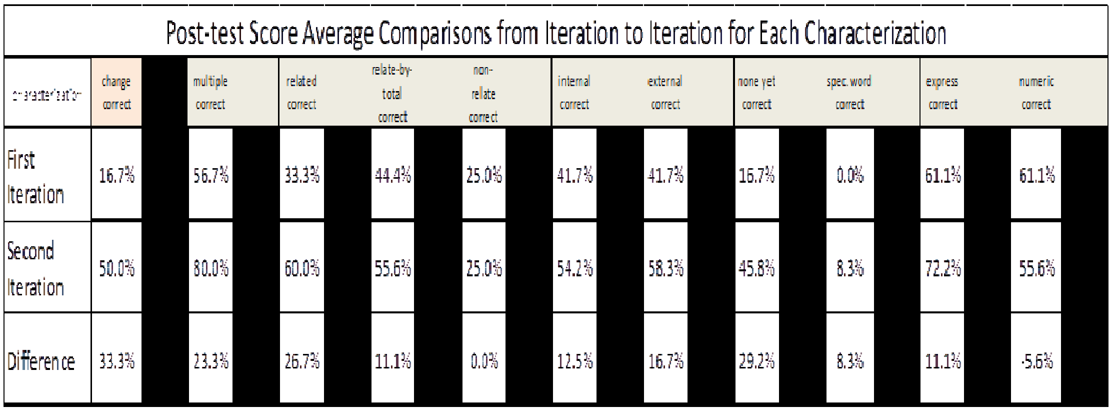
Solution Equation Sources (internally, externally coded) – third characterization. The student must decide whether the problem quantities match, in name and in number, the quantities contained in a pre-existing solution equation (*externally coded*) or no match exists, and the solution equation must be synthesized by the problem-solver from information developed in subsequent steps of the problem-solving process (*internally coded*). First iteration average score for correct identification of a story problem as *internally coded* was 41.7%, and the second iteration average score for the same characterization was 54.2%. The 12.5% difference from one iteration to the next suggests that the modifications made to the pedagogical presentation that supports the student perception of an *internally coded* story problem were reasonably effective but in need of further enhancement to reach the 65% threshold. First iteration average score for correct identification of a story problem as *externally coded* was 41.7%, and the second iteration average score for the same characterization was 58.3%. The 16.7% difference from one iteration to the next suggests that the modifications made to the pedagogical presentation that supports the student perception of an *externally coded* story problem were reasonably effective but in need of further enhancement to reach the 65% threshold.

Solution Equation (special term identification, temporarily unknown) – fourth characterization. If the third characterization (solution equation source) is externally coded, the student must identify the *special term* in the problem text that permitted this choice. If the third characterization was internally coded, the student would temporarily be unable to enter an equation until later in the MSASPT problem-solving process, subsequent to the categorization step. First iteration average score for correctly determining that the *solution equation* for a story problem that has been previously characterized as internally coded cannot be established yet was 16.7%, and the second iteration average score for the same characterization was 45.8%. The 25.2% difference from one iteration to the next suggests that the modifications made to the pedagogical presentation that supports the student perception of a solution equation that must be synthesized from information developed in later steps in the solution process was reasonably effective but in need of further enhancement to reach the 65% threshold.

Both the first iteration and the second iteration average scores for special term identification failed to reach double digits. This aspect of the fourth characterization is dependent almost entirely on the learner's knowledge of the pre-existing solution equations (e.g., area of a rectangle, Pythagorean Theorem, retail sales equation, etc.) that might be found on a GED Mathematics Test. This information would be requisite in passing the test, but not in the general knowledge base of most adults post K-12 education without a concerted effort to familiarize themselves with these solution equations. It was therefore expected that the performance on this characterization would be less than adequate but acceptable at virtually any level knowing that this specific knowledge is algorithmic in nature and acquired when studying to take the Mathematics Test.

Solution Type (expression, numeric) – fifth characterization. The student must decide whether the requested form of the solution is an expression (*expression*) or a number (*numerical*). First iteration average score for correct identification of a story problem as *expression solution type* was 61.1%, and the second iteration average score for the same characterization was 72.2%. The 11.1% difference from one iteration to the next suggests that the modifications made to the pedagogical presentation that supports the student perception of an *expression* solution type story problem were reasonably effective and needed no further enhancement as the performance exceeded the 65% threshold. First iteration average score for correct identification of a story problem as *numerical* solution type was 61.1%, and the second iteration average score for the same characterization was 55.6%. The negative 5.6% difference from one iteration to the next suggests that the modifications made to the pedagogical presentation that supports the student perception of a *numerical* solution type story problem were not effective and the characterization pedagogy needs further enhancement to reach the 65% threshold.

Table 3



Post-test Score Average Comparisons from Iteration to Iteration for Each Characterization

characterization	change correct	multiple correct	related correct	relate-by-total correct	non-relate correct	internal correct	external correct	none yet correct	spec. word correct	express correct	numeric correct
First Iteration	16.7%	56.7%	33.3%	44.4%	25.0%	41.7%	41.7%	16.7%	0.0%	61.1%	61.1%
Second Iteration	50.0%	80.0%	60.0%	55.6%	25.0%	54.2%	58.3%	45.8%	8.3%	72.2%	55.6%
Difference	33.3%	23.3%	26.7%	11.1%	0.0%	12.5%	16.7%	29.2%	8.3%	11.1%	-5.6%

Section 7: Survey Data Findings:

While the survey data was generally very positive with regards to the effectiveness, efficiency, and appeal of the tutoring system, each of the first iteration surveys included use responses to utilities and sub-utilities that the study participant was unable to use due to time constraints. Results of the survey given in terms of the median response for each question in each of the four major teaching utilities over the two iterations can be found in Tables 4-7 below. However, because of the significant number of unqualified responses, these results can only be reflective of the overall acceptance of the tutor as a teaching tool and not as a tool to provide insight as to how the tutor might be changed for the better or whether the mitigations provided relief to the obstacles that were encountered by the participant while using the MS-CPU. Some of the comments made in the free response section corroborate its perceived usefulness:

Subject B1 – "I wish I had more time to work with this program, I would be great to learn more about how to solve story problems."

Subject A1 – "I can see how this will really help people to pass the math test."

Subject B2 – "Ms. Stephens approach to story [problems] was very innovative and explanatory. Her steps took you through each [categorization], and she explained along the way."

Table 4

Survey Question Response Median Data for Main Utility			
Legend: 0-not used; 1-strongly disagree; 2-disagree; 3-no opinion; 4-agree; 5-strongly agree;			
No.	survey question	1st iteration - median score	2nd iteration - median score
1	The basic instructions for how to use the categorization tutor were clear and comprehensive	5	4
2	The instructions for characterizing the first step (quantities) in the categorization process were clear and comprehensive	4	4
3	The instructions for characterizing the second step (relationships) in the categorization process were clear and comprehensive	5	4
4	The instructions for characterizing the third step (solution equation source) in the categorization process were clear and comprehensive	5	4
5	The instructions for characterizing the fourth step (external solution equation presentation) in the categorization process were clear and comprehensive	5	4
6	The instructions for characterizing the fifth step (solution type) in the categorization process were clear and comprehensive	4	4
7	The question mark to the right of each characterization provided a clear and concise (to the point) definition	4	0
8	The hyperlinked word definitions were accurate and concise (to the point)	5	0
9	The hyperlinked word definitions were comprehensive and helped in learning the categorization process	5	0
10	The drag and drop manipulation assisted in learning the categorization process	4	4
11	The drag and drop connection arrows assisted in learning the categorization process	5	3
12	The incompatibilty warnings were helpful in learning the categorization process	5	4
13	The Target Method flow chart for the relationships characterization assisted clearly and concisely (to the point) in understanding this step of the	5	0
14	The code basics (solution equation source) flow chart for the relationships characterization assisted clearly and concisely (to the point) in understanding this step of the categorization process	5	4
15	The check answer utility was accurate and informative and helped in learning the categorization process	5	4
16	The check performance utility was accurate and informative and helped in learning the categorization process	4	4
17	The worked examples utility assisted clearly and comprehensively in the categorization process	5	4
18	The instant progress notification blocks at the bottom of the practice utility were useful in charting my progress from problem to problem	5	4
19	I was able to learn the categorization process from using this practice utility	5	4
20	After using this utility I can project that I believe it will help me in learning to solve many kinds of story problems	5	4

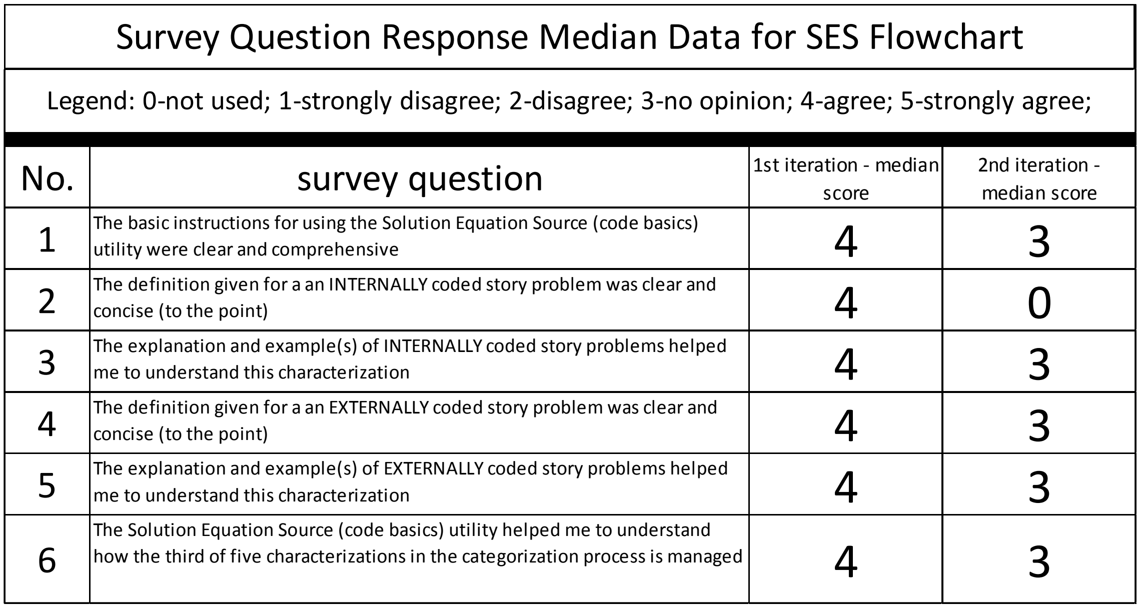
Table 5

Survey Question Response Median Data for TM Flowchart

Legend: 0-not used; 1-strongly disagree; 2-disagree; 3-no opinion; 4-agree; 5-strongly agree;

No.	survey question	1st iteration - median score	2nd iteration - median score
1	The basic instructions for using the Target Method Flow Chart utility were clear and comprehensive	4	1
2	The definition given for a math operation was clear and concise (to the point)	4	3
3	The definition given for the target quantity was clear and concise (to the point)	4	3
4	The definition given for the receiver quantity was clear and concise (to the point)	4	3
5	The presence or lack of these elements in a story problem helped me in determining whether the problem should be characterized as RELATIONAL or not	4	4
6	The definition of a RELATED-BY-TOTAL story problem as a total of two or more quantites given in numerical form helped me to determine this story problem characterization	4	3
7	The definition of a NON-RELATIONAL story problem as an absence of either the 3 elements in a relational problem or of a numerical total helped me to determine this story problem characterization	4	3
8	The Target Method Flow Chart utility helped me to understand how the second of five characterizations in the categorization process is managed	4	3

Table 6



Survey Question Response Median Data for SES Flowchart

Legend: 0-not used; 1-strongly disagree; 2-disagree; 3-no opinion; 4-agree; 5-strongly agree;

No.	survey question	1st iteration - median score	2nd iteration - median score
1	The basic instructions for using the Solution Equation Source (code basics) utility were clear and comprehensive	4	3
2	The definition given for a an INTERNALLY coded story problem was clear and concise (to the point)	4	0
3	The explanation and example(s) of INTERNALLY coded story problems helped me to understand this characterization	4	3
4	The definition given for a an EXTERNALLY coded story problem was clear and concise (to the point)	4	3
5	The explanation and example(s) of EXTERNALLY coded story problems helped me to understand this characterization	4	3
6	The Solution Equation Source (code basics) utility helped me to understand how the third of five characterizations in the categorization process is managed	4	3

Table 7

Survey Question Response Median Data for Worked Examples

Legend: 0-not used; 1-strongly disagree; 2-disagree; 3-no opinion; 4-agree; 5-strongly agree;

No.	survey question	1st iteration - median score	2nd iteration - median score
1	The BASIC INSTRUCTIONS for using the Categorization Worked Examples utility were clear and comprehensive	4	3
2	In the automatic mode (as opposed to drag and drop) the INSTRUCTIONS for EACH STEP were clear and comprehensive	4	4
3	In the automatic mode (as opposed to drag and drop) the REASONS IN SUPPORT FOR EACH CHARACTERIZATION for each step were clear and comprehensive	4	4
4	In the automatic mode (as opposed to drag and drop) all computer behavior was normal and expected	4	3
5	In the drag and drop mode (as opposed to automatic) the INSTRUCTIONS for EACH STEP were clear and comprehensive	4	0
6	In the drag and drop mode (as opposed to automatic) the IMMEDIATE FEEDBACK (correct/incorrect) for EACH STEP was clear and comprehensive	4	0
7	In the drag and drop mode (as opposed to automatic) all computer behavior was normal and expected	4	0
8	I was able to switch between AUTOMATIC and DRAG AND DROP modes of operation while still maintaining the proper categorization sequence	4	0
9	The ability to switch between AUTOMATIC and DRAG AND DROP modes of operation helped me to understand the categorization process	5	0
10	The Worked Examples Categorization utility helped me to understand how the five characterizations determine the completed categorization process	4	4

CHAPTER 4

DISCUSSION

Section 1: Introduction:

The purpose of this formative research study was to discover the elements in the Ms. Stephens' Categorization Practice Utility (MS-CPU) that were problematic for the facilitation of learning how to categorize story problems, so that subsequent studies could be conducted using quantitative methods in conjunction with qualitative methods similar to the ones used in this study to measure the system's teaching effectiveness. Over the course of two study iterations, obstacles to learning were identified and then mitigated according to their location either in the pedagogy or the user interface, producing a tutoring system ready for extended quantitative and qualitative study.

Section 2: Summary of Findings:

Cataloging the various obstacles to an effective, efficient, and appealing tutoring system into four sectors (intuitive use and appeal of the user interface, user interface extraneous cognitive load, categorization pedagogy, and characterization pedagogy) allowed the systematic discovery and mitigation of these obstacles. Identification of the impediments to learning in the user interface by the participant and investigator (observer) resulted in their mitigation according to its required level of design along the continuum of whether the design change must be user-friendly, intuitive, or self-evident (obvious).

Identification of pedagogical impediments to learning originated largely from investigator observation and pre- and post-test performance. There were no apparent impediments in the categorization pedagogy (taxonomy), but each of the characterization pedagogies was found to be impedimental in at least one of three ways.

1. The definition of a term or phrase was unclear.

2. The definition of a characterization term or phrase was inconsistent with the principles or purpose of the taxonomy.

3. The characterization method did not offer a clear algorithmic path to the determination.

The mitigation of each characterization's impediments along these lines was accomplished through clear, concise changes to definitions, either expansion or constriction of characterization determination factors, or the modification or addition of algorithms used to make characterization determinations.

Section 3: Discussion:

The supporting pedagogy for the Ms. Stephens' Algebra Story Problem-solving Tutor (MSASPT) and the subject of this study, the MSASPT Categorization Utility (MS-CPU), derives from the author's hypothesis that difficulty in story problem-solving is the consequence of teaching the domain of story problem-solving using largely conceptual methods. Over several decades of teaching I saw, too often to be coincidental, a sudden drop in learning performance and even in attitude when solution algorithms became solution abstractions or concepts. The prevailing explanation for this fracture was always, "he/she is just not good at math."

Ten years of teaching GED Test Preparation classes, where the mathematics test caused, by far, the most failures in earning a GED Certificate required me to re-think the prevailing explanation. Inductively considering past experiences of my experiences in story problem-solving (they weren't good) and my years of teaching mathematics, including GED, I came to see that the fracture in performance always occurred at the transition from algorithmic solution processes to solutions from the application of mathematical concepts and generalizations. This reasoning led to the conclusion that story problem-solving instruction must be presented algorithmically and as graphically as possible. Only then can the many students apparently stalled at Piaget's Concrete Operational Stage begin to gain the expertise and experience necessary to pass the GED

Mathematics test. Additionally, every sub-process on the way to synthesizing a solution equation must follow this principle.

Presenting the decision-making process for each of the five characterizations that make up a story problem's category as an algorithmic path using multiple approaches (definitions, worked examples, flowcharts) is the hallmark innovation of the MS-CPU system. Such architecture offers a procedural and eventually a conceptual understanding of story problem-solving through successful experience. Also, the MS-CPU tutor extends the utility of story problem categorization to include not only qualifying principles used as unique identifiers for schema retrieval but the collection of information that facilitates problem deep structure recognition, enabling the synthesis of the problem's solution equation.

The MS-CPU pedagogy relies on algorithmic presentations, concretely presented to circumvent the barriers presented by the application of concepts to story problem-solving. In general terms, then, the complex is presented in less complex terms. Likewise, there are algorithmic processes that require their complexity to be presented in less complex terms. Highlighting the components in the problem necessary to traverse the algorithmic process, as alluded to in the Findings chapter (pg. 88), is such a situation. The identification of the words and phrases that were pivotal to making the algorithmic decisions were, in many participant cases, ambiguous.

During the observational sessions of the research, it became obvious that allowing the student to click a button and expose the terms relevant to making characterization decisions was a necessary first step. Essentially "giving the answer" is simply a way in a tutoring system to facilitate not only the continuation of the algorithmic process but a growing schematic collection of what these words and phrases look like for each of the four characterizations that they appear. However, exposing these words and phrases cannot become a permanent crutch in the categorization process, and the use of the buttons is limited. Once the limit is reached, the student is referred to other information (e.g., the Glossary) for more information.

The primary goal of this formative research was to locate then ameliorate design flaws in the MS-CPU tutoring system from a functional perspective to facilitate further, mixed-methods, study. Additional meaning, and perhaps even more consequential, was given to the study when the obstacles indicated by users of the system in the first iteration were mitigated and implemented in the second iteration resulting in higher participant performance levels. Given the hypothesis as described above, this suggests that the conversion to an entirely algorithmic approach for determining the characterizations that constitute the category of a particular story problem and, by extension, story problem-solving itself, did contribute to stronger participant performance. Otherwise, the mitigations performed would have added little or no value to the performances. Further, the increase in pre- to post-test performance for each iteration would make the same suggestion as pre-test characterization instructions were not presented algorithmically while post-test instructions were.

Successful story problem-solving offers itself as the culminating achievement in the study of algebra. A system that offers students and teachers an effective, efficient and appealing path to story problem-solving success can raise levels of achievement in mathematics almost uniformly to this standard. Jaime Escalante once told his mathematics students that "math is the great equalizer." Providing a sense of accomplishment and an avenue for building critical thinking skills as well as higher levels of employment, algebra, with its capstone achievement of story problem solution, is most certainly worth investigating better methods of teaching it.

Other story problem categorization taxonomies or frameworks have been proposed in the last few decades specifically for use in story problem-solving. Two of the most influential for the MS-CPU system were Richard E. Mayer's, "Frequency Norms and Structural Analysis of Algebra Story Problems into Families, Categories, and Templates" in 1981 and Sandra P. Marshall's "Schemas in Problem Solving: An Integrated Model of Learning, Memory, and Instruction" in 1991. Mayer's methods used over a thousand story problems found in ten high school algebra textbooks to construct a framework that consisted of families, categories, and templates.

The synthesis of the MS-CPU taxonomy used his notion of solution equations (or codes) with similar quantities grouped into eight families. The MS-CPU condenses solution formulas into just two families: those that pre-exist the story problem (externally coded) and those that can only be generated by the problem-solver after careful examination of the problem's deep structure and at the same point in time that the problem was presented (internally coded). All solution formulas for externally coded problems that could appear on the GED Mathematics Test are studied previously to the start of the MSASPT (and the MS-CPU).

Marshall's methods presented the concepts of story problem-solving schemas along five categorical lines: (1) Change (2) Group (3) Compare (4) Restate (5) Vary. The elements of her categories were integrated into the MS-CPU pedagogy at various levels. Change, which was predicated on a single quantity's value in a problem changing over time, was contrasted with problems where more than one quantity existed at the same time (concurrently), producing the first characterization in the MS-CPU. Group and compare along with their synonyms were defined as words or phrases that indicated a mathematical operation is to take place and part of the quantity relations characterization definition. Restate became the definition of a relational problem in the quantity relations characterization. Varying was defined as a solution formula (externally coded) for any problem that presented quantity information in the form of a proportion.

The elephant in the room is whether or not the categorization of story problems using MS-CPU will unlock the door to successfully completing the remaining steps in the MSASPT story problem-solving process. Once the effectiveness of the MS-CPU has been optimized, research on the entire MSASPT system can be undertaken.

Section 4: Study Limitations:

In the context of formative research, the number of participants for each iteration was sufficient to accomplish the elimination of the identified instructional obstacles still present after developing the MS-CPU over the period of five years. However, limitations in the study as to the number of iterations, actual time spent using the system, the complexity of the system with its separate intelligent tutoring systems as avenues for learning how to make a characterization decision, may have contributed to an MS-CPU that is only marginally ready for further mixed-methods study. Unfamiliarity with common GED solution formulas limited the characterization evaluation and performance for the fourth characterization.

Section 5: Recommendations for Further Study:

From a mixed-methods approach, continued research would almost certainly include the requirement of a period of at least ten hours with the tutor and a familiarization session (90 minutes) with common external GED solution equations (e.g., area of a triangle, proportions, angular relationships, etc.). For a recommended continuation of formative research iterations, some of the system's complexity and use time requirements can be reduced by eliminating the necessity to use the main categorization practice utility and use the Worked Examples Utility instead. The Worked Examples Utility is a mirror image of the main utility, except for its additional modes of use and the display of characterizations as posted; any obstacles found in its use can be mitigated in both utilities without having to use both. It is expected that one additional study iteration using the Worked Examples Utility will produce a system that would not require investigator coaching and thus be prepared for the mixed-methods research mentioned previously.

Section 6: Conclusions:

The ability to think critically derives first from a willingness to engage in it and second from self-confidence in knowing that it can be done. So much of the environmental, social, and political dysfunction we are presently faced with stems from an inability to think critically. Critical thinking is the gateway to learning how to learn and putting new knowledge to work solving the problems we face. Early in our scholastic careers, we are presented with story problems – the bane of students everywhere. Could it be that an ability to solve story problems systematically and absolutely would work to nurture critical thought in our population, opening the door for a far wider segment of it to productively participate in an enlightened decision-making process? I think so, and it’s certainly worth the effort...

REFERENCES

Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. R. (2009). A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education*, 19(2), 105-154.

Anderson, J. R., Boyle, C. F., & Reiser, B. J. (1985). Intelligent tutoring systems. *Science(Washington)*, 228(4698), 456-462.

Anderson, J.R. (1980). Cognitive psychology and its implications. New York: Freeman.

Arizona Department of Education-Adult Education Services: What Adult Education Means to Arizona (2010) Retrieved from: http://www.azed.gov/adultedservices/files/2014/01/annual-overview-py10-11-final.pdf

Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. *Review of educational research*, 70(2), 181-214.

Atman, K. S. (1987). The role of conation (striving) in the distance education enterprise. *American Journal of Distance Education*, 1(1), 14-24.

Bartlett, F.C. (1932). Remembering: A Study in Experimental and Social Psychology. Cambridge University Press.

Blessing, S. B., & Ross, B. H. (1996). Content effects in problem categorization and problem solving. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22(3), 792.(792)

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13(6), 4-16.

Bobrow, D. G. (1964). Natural language input for a computer problem solving system.

Brain + Cognitive Sciences. Cognitive Science (n.d.) Retrieved from: https://bcs.mit.edu/research/cognitive-science

Bransford, J. D., & Stein, B. S. (1993). The IDEAL problem solver.

Bruner, J. S. (1966). *Toward a theory of instruction* (Vol. 59). Harvard University Press.

Cai, J., & Knuth, E. (Eds.). (2011). *Early algebraization: A global dialogue from multiple perspectives*. Springer Science & Business Media.

Cai, J., Lew, H. C., Morris, A., Moyer, J. C., Ng, S. F., & Schmittau, J. (2005). The development of student's' algebraic thinking in earlier grades. *Zentralblatt für Didaktik der Mathematik*, 37(1), 5-15.

Carbonell, J. G. (1981, August). A computational model of analogical problem solving. In *IJCAI* (Vol. 81, pp. 147-152).

Catrambone, R., & Holyoak, K. J. (1989). Overcoming contextual limitations on problem-solving transfer. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 15(6), 1147.

Chandler, P.; Sweller, J. (1991). "Cognitive Load Theory and the Format of Instruction". Cognition and Instruction. 8 (4): 293–332. doi:10.1207/s1532690xci0804_2

Cheong, Y. K., & Consultancy, M. (2002). The model method in Singapore. *The Mathematics Educator*, 6(2), 47-64.

Chi, M. T., Feltovich, P. J., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive science*, 5(2), 121-152.

Chi, M., & VanLehn, K. (2007). Accelerated future learning via explicit instruction of a problem solving strategy. FRONTIERS IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS, 158, 409. (411)

Chris Hedges, "My Teacher", 7/19/2015 retrieved from: Cognitive, Affective, and Psychomotor Domains – Grading (n.d.). Retrieved from: http://courses.washington.edu/pharm439/Bloomstax.htm

Corbett, A., Wagner, A. Z., Lesgold, S., Ulrich, H., & Stevens, S. M. (2006). *The impact on learning of generating vs. selecting descriptions in analyzing algebra example solutions*. In S. A. Barab, K. E. Hay, & D. T. Hickey (Eds.), The 7th International Conference of the Learning Sciences (pp. 99-105). Mahwah, NJ: Erlbaum.

Critical Issues in Mathematics Education Series, Volume 5, Teaching Undergraduates Mathematics, Berkeley, CA, 2009. Berkeley: Mathematical Sciences Research Institute.

Davidson, J. E. & Sternberg, R. J. (1998). Smart problem solving: How meta-cognition helps. In D. J. Hacker, J. Dunlosky, & A. C. Graesser (Eds.), Metacognition in educational theory and practice (pp. 47-68). Mahwah, NJ: Lawrence Erlbaum.

Derry, S. J. and the TiPS Research Group (2001). Development and Assessment of Tutorials in Problem Solving (TiPS): A Remedial Mathematics Tutor. Final Report to the Office of Naval Research (N00014-93-1-0310), Wisconsin Center for Education Research, University of Wisconsin-Madison, Madison, WI.

Drejing, K. (2013). Abstractions in Judgment: Does Construal Level Influence the System Specific Reasoning of Dual Process Theory?.

Dubow, E. F., Boxer, P., & Huesmann, L. R. (2009). Long-term effects of parents' education on children's educational and occupational success: Mediation by family interactions, child aggression, and teenage aspirations. *Merrill-Palmer quarterly (Wayne State University. Press)*, 55(3), 224.

Edgar, D. W. (2012). Learning Theories and Historical Events Affecting Instructional Design in Education. *Sage Open*, 2(4), 2158244012462707.

Esty, W. W., & Teppo, A. R. (1996). Algebraic thinking, language and word problems. In P. C. Elliot & M. J. Kenney (Eds.), *Communication in Mathematics, K-12 and Beyond* (pp.45-53). Reston, VA.: 1996 Yearbook. National Council of Teachers of Mathematics, Inc.

Eysenck, M. W., & Keane, M. T. (2000). *Cognitive psychology: A student's handbook*. Taylor & Francis.

Flavell, J. H. (1976). Metacognitive aspects of problem solving. *The nature of intelligence*, *12*, 231-235.

Gagne, R. M. (1980). Is educational technology in phase?. *Educational Technology*.

Gagne, R. M. & Briggs, L. J.. (1979) Principles of Instructional Design, 2nd ed. New York: Holt, Rinehart and Winston.

GED Testing: Research; *2013 GED Testing Program Statistical Report(2014)*. Retrieved from: http://www.gedtestingservice.com/educators/historical-testing-data

Gerjets, P., Scheiter, K., Catrambone, R. (2004) Designing Instructional Examples to Reduce Intrinsic Cognitive Load: Molar versus Modular Presentation of Solution Procedures. *Instructional Science*, 32 (1-2), (pp.33-58). Springer Verlag.

Gick, M.L., & Holyoak, K.J. (1983). Schema induction and analogical transfer*. Cognitive Psychology*, 15, 1-38.

Harris, R. A. (2004). Voice interaction design: crafting the new conversational speech systems. Elsevier.

Heffernan, N. T., & Koedinger, K. R. (1997). The composition effect in symbolizing: The role of symbol production vs. text comprehension. In Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society (pp. 307-312). Hillsdale, NJ: Lawrence Erlbaum Associates.

Hegarty, M., Mayer, R. E., & Monk, C. A. (1995). Comprehension of arithmetic word problems: A comparison of successful and unsuccessful problem solvers. Journal of educational psychology, 87(1), 18.

Hickman, J. A. Sequential and Simultaneous Cognitive Processing. *Encyclopedia of Special Education*.

Hiebert, J., & Lefevre, P. (1986). Conceptual and procedural knowledge in mathematics: An introductory analysis. In J. Hiebert (Ed.), Conceptual and procedural knowledge: The case of mathematics (pp. 1-27). Hillsdale, NJ: Lawrence Erlbaum Associates.

Hilbert, T. S., Renkl, A., Schworm, S., Kessler, S., & Reiss, K. (2008). Learning to teach with worked-out examples: a computer-based learning environment for teachers. *Journal of computer assisted learning*, 24(4), 316-332.

Hohn, R. L., & Frey, B. (2002). Heuristic training and performance in elementary mathematical problem solving. *The Journal of Educational Research*, 95(6), 374-380. (375)

Holyoak, K. J., & Thagard, P. R. (1989). A computational model of analogical problem solving. *Similarity and analogical reasoning*, 242-266.(242)

Holyoak, K.J. (1995) "Chapter 8; Problem Solving." In An Invitation to Cognitive Science, 2nd ed., 3:267–96. The MIT Press.

Hong, N. S. (1998). The relationship between well-structured and ill-structured problem solving in multimedia simulation (Doctoral dissertation, The Pennsylvania State University).

Hoth, W. E. (1961). From Skinner to Crowder to Chance: A primer on teaching machines. *The English Journal*, *50*(6), 398-401.

It's Time to Give Noel Burch Some Credit. (n.d.). Retrieved from http://exceptionalleaderslab.com/its-time-to-give-noel-burch-some-credit/

Jitendra, A. K., Griffin, C., Deatline-Buchman, A., Dipipi-Hoy, C., Sczesniak, E., Sokol, N. G., & Xin, Y. P. (2005). Adherence to mathematics professional standards and instructional design criteria for problem-solving in mathematics. *Exceptional Children*, 71(3), 319-337.

Jonassen, D. "Let Us Learn to Solve Problems - Paper for Discussion on ITFORUM," 2005. itforum.coe.uga.edu/paper83/paper83.html.

Jonassen, D. H. (2003). Designing research-based instruction for story problems. *Educational Psychology Review*, 15(3), (12).

Jonassen, D. H. (2004). Learning to solve problems: An instructional design guide (Vol. 6). John Wiley & Sons.

Jonassen, D. H. (2010). Learning to solve problems: A handbook for designing problem-solving learning environments. Routledge.

Jonassen, D.H. & Tessmer, M. (1996/1997). An outcomes-based taxonomy for instructional systems design, evaluation, and research*. Training Research Journal,* 2, 11-46.

Jonassen, D. H., Mayes, T. & McAleese, R. (1993). A manifesto for a constructivist approach to uses of technology in higher education. In Duffy, T. M., Lowyck, J. & Jonassen, D. H. (Eds.) *Designing environments for constructive learning* (pp. 231-247). Berlin; Springer-Verlag.

Jonassen, D. H. (2000) "Toward a Design Theory of Problem Solving." *Educational Technology Research and Development* 48, no. 4: 63–85.

Jonnasen, D., (n/n) Toward a Meta-Theory of Problem Solving. Retrieved from: http://web.missouri.edu/jonassend/problems.htm

Kintsch, W. (1977). On comprehending stories. *Cognitive processes in comprehension.* Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Kintsch, W. (1988). The role of knowledge in discourse comprehension: a construction-integration model. *Psychological review*, 95(2), 163.

Kintsch, W., & Greeno, J. G. (1985). Understanding and solving word arithmetic problems. *Psychological review*, *92*(1), 109.Kintsch, 1988.

Kintsch, W., & Van Dijk, T. A. (1978). Toward a model of text comprehension and production. *Psychological review*, *85*(5), 363.

Kitchener, K. S., King, P. M., Wood, P. K., & Davison, M. L. (1989). Sequentiality and consistency in the development of reflective judgment: A six-year longitudinal study. *Journal of Applied Developmental Psychology,10*(1), 73-95.

Koedinger, K. R., & Corbett, A. (2006). Cognitive tutors. *In K. Sawyer (Ed.), The Cambridge Handbook of the Learning Sciences, Cambridge University Press*, 61-78.

Kodaganallur, V., Weitz, R. R., & Rosenthal, D. (2005). A comparison of model-tracing and constraint-based intelligent tutoring paradigms. International Journal of Artificial Intelligence in Education, 15(2), 117-144.

Kozhevnikov, M. (2007). Cognitive styles in the context of modern psychology: toward an integrated framework of cognitive style. *Psychological bulletin*, 133(3), 464.

Krathwohl, D. R., Bloom, B. S., & Masia, B. B. (1973). Taxonomy of Educational Objectives, Handbook I: Affective Domain, David McKay, New York. *Krathwohl Taxonomy of Educational Objectives*1973.

Kuhn, D., Garcia-Mila, M., Zohar, A., Andersen, C., White, S. H., Klahr, D., & Carver, S. M. (1995). Strategies of knowledge acquisition. *Monographs of the society for research in child development*, i-157.

Laurillard, D. (1997). Styles and approaches in problem-solving. *The experience of learning*, 2. (127)

Lawler, R. W. (1981). The progressive construction of mind. Cognitive Science, 5, 1-30.

Lester, F. K. (1994). Musings about mathematical problem-solving research: 1970-1994. *Journal for research in mathematics education*, 25(6), 660-675.

Livingston, J. A. (2003). Metacognition: An Overview.

Marcus Annaeus Seneca, Roman philosopher, ca. 4 BC–AD 65

Marshak, R. (2010) *User Interfaces Shouldn't Just Be Intuitive*. Retrieved from: http://www.customers.com/articles/user-interfaces-shouldnt-just-be-intuitive/

Marshall, S. P. (1992). Schemas in Problem Solving: An Integrated Model of Learning, Memory, and Instruction (No. CRSE-91-02). SAN DIEGO STATE UNIV CA CENTER FOR RESEARCH IN MATHEMATICS AND SCIENCE EDUCATION.

Marshall, S. P. (1995). *Schemas in problem solving*. Cambridge University Press.

Masui, C., & De Corte, E. (1999). Enhancing learning and problem solving skills: orienting and self-judging, two powerful and trainable learning tools. *Learning and instruction*, *9*(6), 517-542.

Mayer, R. E. (1981). Frequency norms and structural analysis of algebra story problems into families, categories, and templates. Instructional Science, 10(2), 135-175.

Mathews, J. (1988). Escalante: The best teacher in America. Holt.

Mayer, R. E. (1988). Learning strategies: An overview. *Learning and study strategies: Issues in assessment, instruction, and evaluation*, 11-22.

Mayer, R. E. (2001). Cognitive, metacognitive, and motivational aspects of problem solving. In *Metacognition in learning and instruction* (pp. 87-101). Springer Netherlands.

Mayer, R. E. (2002). Rote versus meaningful learning. *Theory into practice*, *41*(4), 226-232.McClelland, 2009.

McCorduck, Pamela (2004), Machines Who Think (2nd ed.), Natick, MA: A. K. Peters, Ltd., ISBN 1-56881-205-1, pp. 161–170.

Merrill, D. C., Reiser, B. J., Ranney, M., & Trafton, J. G. (1992). Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences*, *2*(3), 277-305.

Messick, S. (1984). The nature of cognitive styles: Problems and promise in educational practice. *Educational psychologist*, *19*(2), 59-74.

Minsky, M. (1975, June). Minsky's frame system theory. In *Theoretical issues in natural language processing* (pp. 104-116).

Mitrovic, A., Mayo, M., Suraweera, P., & Martin, B. (2001, June). Constraint-based tutors: a success story. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (pp. 931-940). Springer Berlin Heidelberg.

Murray, R. C., VanLehn, K., & Mostow, J. (2004). Looking ahead to select tutorial actions: A decision-theoretic approach. *International Journal of Artificial Intelligence in Education*, 14(3, 4), 235-278.

Nathan, M. J., Kintsch, W., & Young, E. (1992). A theory of algebra-word-problem comprehension and its implications for the design of learning environments. *Cognition and instruction*, 9(4), 329-389.

National Council of Teachers of Mathematics (1980) Retrieved from: http://www.nctm.org/Research-and-Advocacy/research-brief-and-clips/Problem-Solving/

National Council of Teachers of Mathematics. (1989). Curriculum and evaluation standards for school mathematics. Reston, VA: Wilson, J. W., Fernandez, M. L., & Hadaway, N. (1993). Mathematical problem solving. *Research ideas for the classroom: High school mathematics*, 57-78.

National Education Goals Panel. (1997). The National Education Goals report: Building a nation of learners, 1997. Washington, DC: U.S. Government Printing Office.

Nielsen, J., (1995) How to Conduct a Heuristic Evaluation. Retrieved from http://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/

Newell, A., & Simon, H. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice Hall.

Newell, A., Shaw, J. C., & Simon, H. A. (1958). Elements of a theory of human problem solving. *Psychological review*, 65(3), 151.(151)

Ng, S. F., & Lee, K. (2009). The model method: Singapore children's tool for representing and solving algebraic word problems. *Journal for Research in Mathematics Education*, 282-313.(283)

Nilsson, N. J. (2009). The quest for artificial intelligence. Cambridge University Press.

Novick, L. R. (1988). Analogical transfer, problem similarity, and expertise. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 14(3), 510.

Novick, L. R., & Holyoak, K. J. (1991). Mathematical problem solving by analogy. *Journal of experimental psychology: Learning, memory, and cognition*, 17(3), 398.

Novotna, J. (2001). [Review of the book Making Sense of Word Problems, by L. Verschaffel, B. Greer & E. deCorte]. Contexts of Learning (Classroom, Schools, and Society), 8.

Nwana, H. S. (1990). Intelligent tutoring systems: an overview. *Artificial Intelligence Review*, 4(4), 251-277.

Ohlsson, S. (1992). Information-processing explanations of insight and related phenomena. *Advances in the psychology of thinking*, chap-1.

Pane, J. F., Griffin, B. A., McCaffrey, D. F., & Karam, R. (2013). Effectiveness of cognitive tutor algebra I at scale. *Educational Evaluation and Policy Analysis*, 0162373713507480.

Pithers, R. T. (2006). Cognitive learning style: A review of the field dependent-field independent approach. *Journal of Vocational Education and Training*.

Polya, G. (1945) How to solve it. Princeton, NJ, Princeton University Press.

Post, T., & Cramer, K. (1989). Knowledge, Representation and Quantitative Thinking. In M. Reynolds (Ed.) Knowledge Base for the Beginning Teacher - Special publication of the AACTE (pp. 221-231). Oxford: Pergamon Press.

Richgels, D. (1984). Dispelling the mystery about comprehension: Kintch's model and implications for instruction. *Reading Horizons*, 24(3), (173)

Roberts, M. & Erdos, G. (1993) Strategy Selection and Metacognition, Educational Psychology, 13:3-4, 259-266, DOI: 10.1080/0144341930130304

Roedinger, H. (2004). What Happened to Behaviorism. Association for Psychological Science. Retrieved from: http://www.psychologicalscience.org/index.php/uncategorized/what-happened-to-behaviorism.html.

Rosch, E. (1978). Principles of Categorization. In E. Roach & B. Lloyd (Eds.), Cognition and Categorization (pp. 27-48). Hillsdale, New Jersey: Lawrence Erlbaum Associates, Publishers.

Rumelhart, D. E. (1978). *Schemata: The building blocks of cognition*. Center for Human Information Processing, University of California, San Diego.

Rumelhart, D. E. (1991). Understanding understanding. *Memories, thoughts and emotions: Essays in honor of George Mandler*, 257-275.

Scheiter, K., Gerjets, P., & Catrambone, R. (2006). Making the abstract concrete: Visualizing mathematical solution procedures. *Computers in Human Behavior*, 22(1), 9-25.

Schoenfeld, A. H. (1992). Learning to think mathematically: Problem solving, metacognition, and sense making in mathematics. *Handbook of research on mathematics teaching and learning*, 334-370.(335)

Schustack, M. W., & Anderson, J. R. (1979). Effects of analogy to prior knowledge on memory for new information. *Journal of Verbal Learning and Verbal Behavior*, 18(5), 565-583.

Shavelson, R. J. (1972). Some aspects of the correspondence between content structure and cognitive structure in physics instruction. *Journal of educational psychology*, 63(3), 225.

Simon, H. A., & Newell, A. (1971). Human problem solving: The state of the theory in 1970. *American Psychologist*, 26(2), 145. (145).

Skinner, B. F. (1954). The science of learning and the art of teaching. *Cambridge, Mass, USA*, 99-113.

Skinner, B. F. (1958). Teaching machines. *Science*.

Smith, E. E., Patalano, A. L., & Jonides, J. (1998). Alternative strategies of Categorization. Cognition, 65(2), 167-196.

Smith, E. E., & Sloman, S. A. (1994). Similarity-versus rule-based categorization. Memory & Cognition, 22(4), 377-386.

Smith, M.U. (1991). A view from biology. In M.U. Smith (ed.), Toward a unified theory of problem solving. Hillsdale, NJ: Lawrence Erlbaum Associates.

Sternberg, R., & Sternberg, K. (2016). *Cognitive psychology*. Nelson Education.

Stewin, L., & Anderson, C. (1974). Cognitive complexity as a determinant of information processing*. Alberta Journal of Educational Research,* 20(3), 233-243.

Stigler, J. W., Fuson, K. C., Ham, M., & Sook Kim, M. (1986). An analysis of addition and subtraction word problems in American and Soviet elementary mathematics textbooks. *Cognition and instruction*, 3(3), 153-171.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science,* 12, 257-285.

Sweller, J., & Cooper, G. A. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction*, 2(1), 59-89.

The History of Educational Technology (n.d.) retrieved from: http://people.ischool.illinois.edu/~chip/projects/timeline/1969an.html.

Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological review*, 55(4), 189.

Uhr, L., & Kochen, M. (1969, May). MIKROKOSMS and robots. In *Proceedings of the 1st international joint conference on Artificial intelligence* (pp. 541-555). Morgan Kaufmann Publishers Inc.

Uhr, L., & Vossler, C. (1961). A pattern recognition program that generates, evaluates, and adjusts its own operators. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference* (pp. 555-569). ACM.

VanLehn, K. (1988). *Problem solving and cognitive skill acquisition* (No. AIP-32). Carnegie-Mellon Univ. Pittsburgh PA Artificial Intelligence and Psychology Project.(4)

VanLehn, K. (2006). The behavior of tutoring systems. *International journal of artificial intelligence in education*, 16(3), 227-265.

VanLehn, K. (2011a). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. Educational Psychologist, 46(4), 197-221.

VanLehn, K. (2011b). Lecture on Structure of Tutoring Systems. Personal Collection of K. VanLehn, Arizona State University, Tempe, AZ.

VanLehn, K., Bhembe, D., Chi, M., Lynch, C., Schulze, K., Shelby, R., ... & Wintersgill, M. (2004, August). Implicit versus explicit learning of strategies in a non-procedural cognitive skill. In *International Conference on Intelligent Tutoring Systems* (pp. 521-530). Springer Berlin Heidelberg.

VanLehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R., Taylor, L., ... & Wintersgill, M. (2005). The Andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence in Education*, 15(3), 147-204.

Vygotsky, L. (1978). Interaction between learning and development. *Readings on the development of children*, 23(3), 34-41.

Wachtel, P. L. (1980). Transference, schema, and assimilation. *Annua Psychoanal*, *8*, 59-76.

Wang, Y., & Chiew, V. (2010). On the cognitive process of human problem solving. Cognitive Systems Research, 11(1), 81-92. (82)

Ward, M., & Sweller, J. (1990). Structuring effective worked examples. *Cognition and instruction*, 7(1), 1-39.

Watson, J. B. (1994). Psychology as the behaviorist views it. Psychological Review, 101(2), 248-253. Retrieved from: http://login.ezproxy1.lib.asu.edu/login?url=http://search.proquest.com/docview/614324198?accountid=4485

Wilson, J. W., Fernandez, M. L., & Hadaway, N. (1993). Mathematical problem solving. *Research ideas for the classroom: High school mathematics*, 57-78.

Winston, P. H. (1980). Learning and reasoning by analogy. *Communications of the ACM*, 23(12), 689-703.

Witkin, H. A., Moore, C. A., Goodenough, D. R., & Cox, P. W. (1975). Field-dependent and field-independent cognitive styles and their educational implications. *ETS Research Bulletin Series*, 1975(2), 1-64.

Xin, Y. P. (2008). The effect of schema-based instruction in solving mathematics word problems: An emphasis on prealgebraic conceptualization of multiplicative relations. *Journal for Research in Mathematics Education*, 526-551.

Yeap, B. H., & Kaur, B. (2001). Semantic characteristics that make arithmetic word problems difficult. In Proceedings of the 24th Annual Conference of the Mathematics Education Research Group of Australasia Incorporated (MERGA) on "Numeracy and Beyond.

Yolles, M. (2006). Organizations as complex systems: An introduction to knowledge cybernetics (Vol. 2). IAP.

Young, R. M. (2001). Production systems in cognitive psychology. *International encyclopedia of the social and behavioral sciences*, 12-143.

APPENDIX A

INVESTIGATOR OBSERVATION GUIDELINES

Observation Guidelines - Overall System	
Subject comprehension of instructions	
Subject comprehension of definitions	
Categorization Flow	
User interface design - intuitiveness	
Miscellaneous observations	

APPENDIX B

CATEGORIZATION PRE-TEST

MSASPT Categorization Pre-test

1	Draymond's farm has a fenced yard that holds goats and chickens. If the total number of animal feet in the yard is 52 and the total number of animals is 16, how many goats are in the yard?										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	none yet	special word	express.	numeric
	o	o	o	o	o	o	o	o		o	o
2	Charley owns a free-range egg farm. He has Y chickens now, having lost 7 of them to coyotes this past week. Write an expression for the number of chickens he had before the coyotes struck.										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	none yet	special word	express.	numeric
	o	o	o	o	o	o	o	o		o	o
3	The area of a rectangle is A sq. units. If its length is four times its width write an expression that represents its width.										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	none yet	special word	express.	numeric
	o	o	o	o	o	o	o	o		o	o
4	Georgia is five years less than four times as old as Vicki. If the difference between their ages is 16, how old is Georgia?										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	none yet	special word	express.	numeric
	o	o	o	o	o	o	o	o		o	o
5	In 11 years Erica will be 5 years less than 3 times as old as she is now. How old is she now?										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	none yet	special word	express.	numeric
	o	o	o	o	o	o	o	o		o	o
6	Samantha has $8 more than twice as much money to spend at Disneyland than Portia. Gina has $8 less to spend than Samantha.Write an expression that represents how much Gina can spend at Disneyland.										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	none yet	special word	express.	numeric
	o	o	o	o	o	o	o	o		o	o

7	Rena wants to re-tile her kitchen floor. The tile costs $1.10 per square foot including tax and the labor to install it is $.20 per square foot. Write an expression that represents the cost of her new tile floor.										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	none yet	special word	express.	numeric
	o	o	o	o	o	o	o	o		o	o
8	Theodore wants to build a bridge across his backyard pool. He has determined that he will need 200 feet of 2" x 4" redwood lumber that is old in 8' lengths. If each 2"x 4"x 8' costs $3.50 and the tax is 10% how much will Theodore pay for the lumber?										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	none yet	special word	express.	numeric
	o	o	o	o	o	o	o	o		o	o
9	A city planner is laying out the boundaries of a new small triangular skateboard park in the center of a larger park. If the area of the small triangular park can be no more than 200 sq. ft. and its base no more than 25 ft. what is its maximum height?										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	none yet	special word	express.	numeric
	o	o	o	o	o	o	o	o		o	o
10	A yard stick is broken into two pieces. If one piece of the yard stick is Y inches long, write an expression for the length of the other piece in inches. You need not include the unit, inches, in your answer. A yardstick is 36 inches long.										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	none yet	special word	express.	numeric
	o	o	o	o	o	o	o	o		o	o
11	A rectangle's length is 3 units less than 3 times it width. If the area of the rectangle is 60 square units what must be its length?										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	none yet	special word	express.	numeric
	o	o	o	o	o	o	o	o		o	o
12	A car is traveling at a speed of Y miles per hour. Write an expression that represents how long it will take the driver to get from mile marker 82 to mile marker 102 on I-10.										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	none yet	special word	express.	numeric
	o	o	o	o	o	o	o	o		o	o

APPENDIX C

CATEGORIZATION POST-TEST

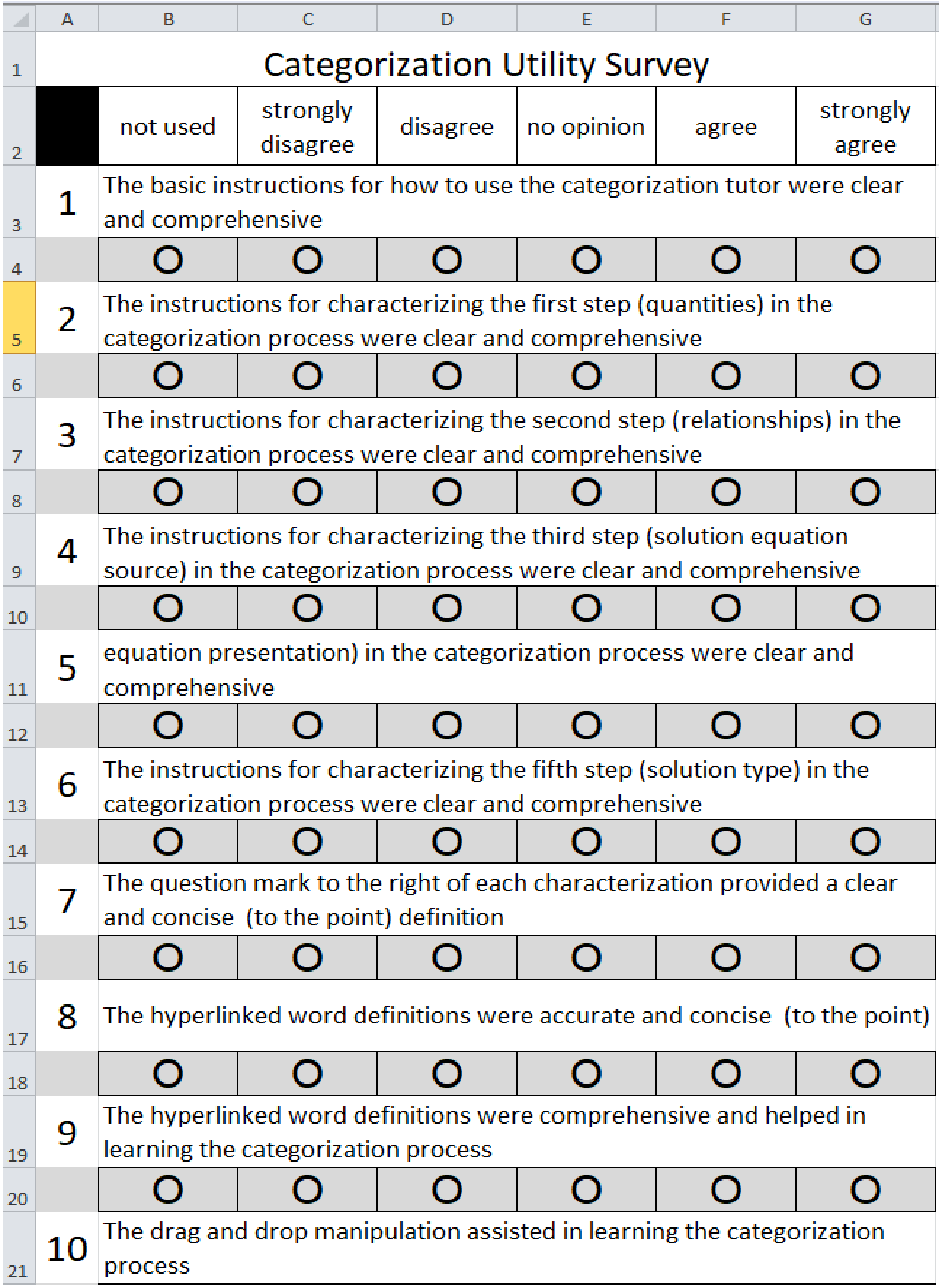
MSASPT Categorization Post-test

1	The difference between a rectangle's length and its width is 7 units. Choose one of the two distances to be the independent variable and write an expression for the dependent quantity. Remember that, in a rectangle, the length is always greater than the width.										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	none yet	special word	express.	numeric
	O	O	O	O	O	O	O	O		O	O
2	Jenny is an auto mechanic. She was cleaning up her work station and noticed that the difference between the number of bolts she found and the number of nuts was 23. If there are 9 more than twice as many nuts as bolts, how many nuts did she find?										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	none yet	special word	express.	numeric
	O	O	O	O	O	O	O	O		O	O
3	In a recent election 35% of the voters voted for the Republican candidate and Y% voted for the Democratic candidate. The Green Party candidate received a significant number of votes as well. Write an expression for the percent of voters who voted Green Party. There were no write-in candidates. Remember, the total percent of votes is 100%.										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	none yet	special word	express.	numeric
	O	O	O	O	O	O	O	O		O	O
4	Olga collects baseball cards. Three months from now she expects to have 3 less than 5 times as many cards as she has now. If she has Y cards now, write an expression that represents the number of cards that she will have in three months.										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	none yet	special word	express.	numeric
	O	O	O	O	O	O	O	O		O	O
5	A boat is traveling at a speed of 60 miles per hour. Write an expression that represents how long it will take for the boat to go from Point Reyes to the San Francisco Bay Bridge, a distance of Y miles.										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	none yet	special word	express.	numeric
	O	O	O	O	O	O	O	O		O	O
6	If Arnold buys two hot dogs and two drinks for he and his son, write an expression that represents his cost if each hot dog was $1.50 and each drink was $Y.										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	none yet	special word	express.	numeric
	O	O	O	O	O	O	O	O		O	O

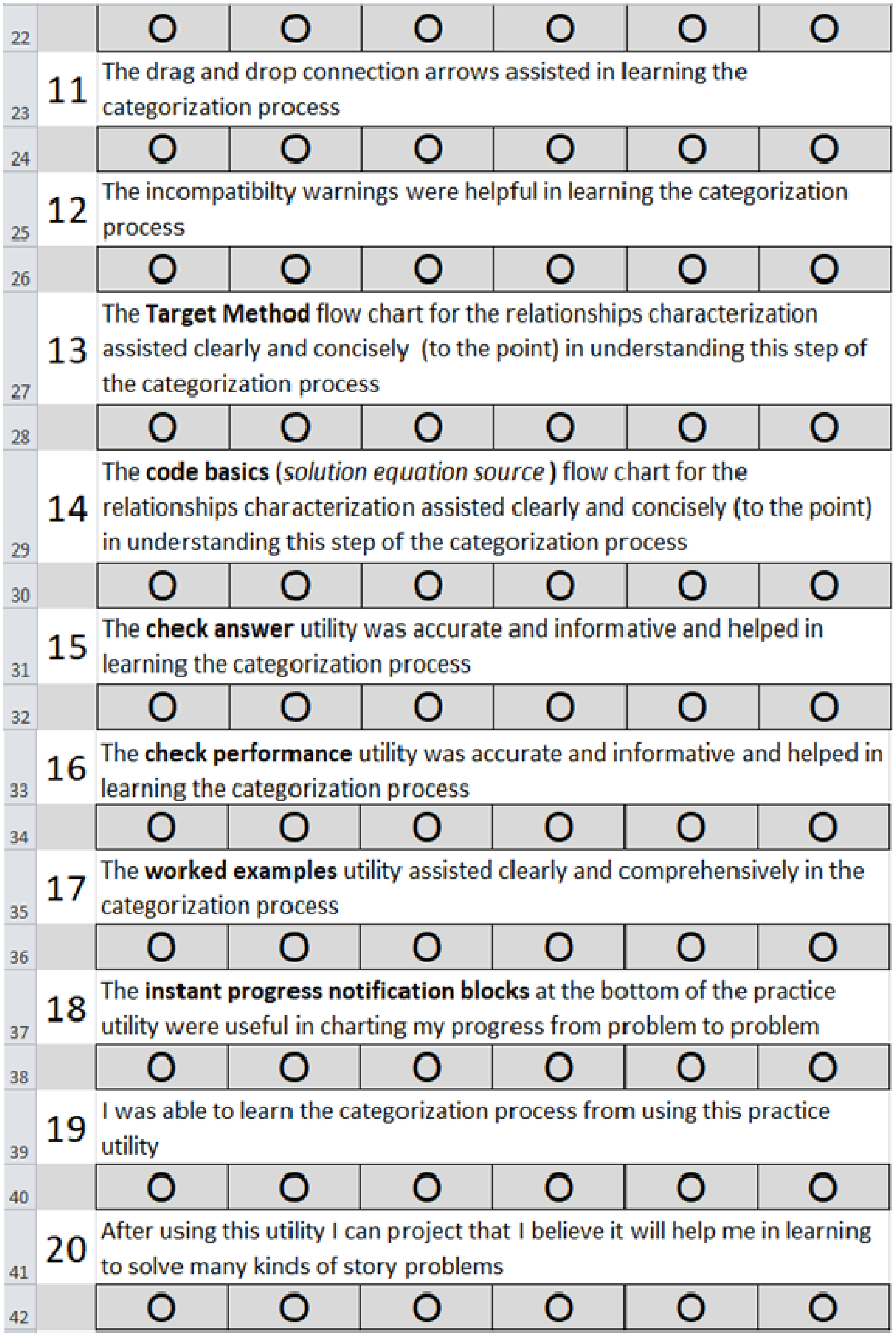
7	If the base of an isosceles triangle is 5 units less than 3 times one of its sides write an expression that represents its perimeter.										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	noneyet	special word	express.	numeric
	0	0	0	0	0	0	0	0		0	0
8	If there are 7 more than 3 times as many dogs as cats in a pet hotel and there are 43 pets altogether, how many dogs are in the hotel?										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	noneyet	special word	express.	numeric
	0	0	0	0	0	0	0	0		0	0
9	James's stock holdings have increased in value $400 less than 3 times as much as he started with. If his holdings are now worth $7,100 what was the original value of his holdings?										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	noneyet	special word	express.	numeric
	0	0	0	0	0	0	0	0		0	0
10	Leandro rents a car for an initial payment of $19.89 and $.20 for every mile that he drives the car. Returning the car after filling it up with gas the odometer shows that he has driven 195 miles. If the total cost,including the fill-up, was $78.00 how much did he pay for the fill-up?										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	noneyet	special word	express.	numeric
	0	0	0	0	0	0	0	0		0	0
11	Fiorina wants to carpet both her living room rooms. The living room is 12' by 20'. The family room is right next to the living room in the shape of a half-circle whose diameter is equal to the width of the living room. The carpet is $1.98 a square foot; how which will the carpet cost?										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	noneyet	special word	express.	numeric
	0	0	0	0	0	0	0	0		0	0
12	The hypotenuse of a right triangle measures 25 inches. If leg A, of this triangle, is 5 inches more than leg B, what must be the measure of leg A?										
	quantity number		relationships			soln code source		solution code		solution type	
	change	multiple	related	rel. by total	non-related	internal	external	noneyet	special word	express.	numeric
	0	0	0	0	0	0	0	0		0	0

APPENDIX D

PARTICIPANT SURVEY FORM – MAIN CATEGORIZATION UTILITY



	A	B	C	D	E	F	G
1	Categorization Utility Survey						
2		not used	strongly disagree	disagree	no opinion	agree	strongly agree
3	1	The basic instructions for how to use the categorization tutor were clear and comprehensive					
4		○	○	○	○	○	○
5	2	The instructions for characterizing the first step (quantities) in the categorization process were clear and comprehensive					
6		○	○	○	○	○	○
7	3	The instructions for characterizing the second step (relationships) in the categorization process were clear and comprehensive					
8		○	○	○	○	○	○
9	4	The instructions for characterizing the third step (solution equation source) in the categorization process were clear and comprehensive					
10		○	○	○	○	○	○
11	5	equation presentation) in the categorization process were clear and comprehensive					
12		○	○	○	○	○	○
13	6	The instructions for characterizing the fifth step (solution type) in the categorization process were clear and comprehensive					
14		○	○	○	○	○	○
15	7	The question mark to the right of each characterization provided a clear and concise (to the point) definition					
16		○	○	○	○	○	○
17	8	The hyperlinked word definitions were accurate and concise (to the point)					
18		○	○	○	○	○	○
19	9	The hyperlinked word definitions were comprehensive and helped in learning the categorization process					
20		○	○	○	○	○	○
21	10	The drag and drop manipulation assisted in learning the categorization process					



	○	○	○	○	○	○
11	The drag and drop connection arrows assisted in learning the categorization process					
	○	○	○	○	○	○
12	The incompatibilty warnings were helpful in learning the categorization process					
	○	○	○	○	○	○
13	The **Target Method** flow chart for the relationships characterization assisted clearly and concisely (to the point) in understanding this step of the categorization process					
	○	○	○	○	○	○
14	The **code basics** (*solution equation source* **)** flow chart for the relationships characterization assisted clearly and concisely (to the point) in understanding this step of the categorization process					
	○	○	○	○	○	○
15	The **check answer** utility was accurate and informative and helped in learning the categorization process					
	○	○	○	○	○	○
16	The **check performance** utility was accurate and informative and helped in learning the categorization process					
	○	○	○	○	○	○
17	The **worked examples** utility assisted clearly and comprehensively in the categorization process					
	○	○	○	○	○	○
18	The **instant progress notification blocks** at the bottom of the practice utility were useful in charting my progress from problem to problem					
	○	○	○	○	○	○
19	I was able to learn the categorization process from using this practice utility					
	○	○	○	○	○	○
20	After using this utility I can project that I believe it will help me in learning to solve many kinds of story problems					
	○	○	○	○	○	○

APPENDIX E

PARTICIPANT SURVEY FORM – TARGET METHOD FLOWCHART

TM Flow Chart Survey

	not used	strongly disagree	disagree	no opinion	agree	strongly agree
1	The basic instructions for using the Target Method Flow Chart utility were clear and comprehensive					
	O	O	O	O	O	O
2	The definition given for a math operation was clear and concise (to the point)					
	O	O	O	O	O	O
3	The definition given for the target quantity was clear and concise (to the point)					
	O	O	O	O	O	O
4	The definition given for the receiver quantity was clear and concise (to the point)					
	O	O	O	O	O	O
5	The presence or lack of these elements in a story problem helped me in determining whether the problem should be characterized as RELATIONAL or not					
	O	O	O	O	O	O
6	The definition of a RELATED-BY-TOTAL story problem as a total of two or more quantites given in numerical form helped me to determine this story problem characterization					
	O	O	O	O	O	O
7	The definition of a NON-RELATIONAL story problem as an absence of either the 3 elements in a relational problem or of a numerical total helped me to determine this story problem characterization					
	O	O	O	O	O	O
8	The Target Method Flow Chart utility helped me to understand how the second of five characterizations in the categorization process is managed					
	O	O	O	O	O	O

APPENDIX F

PARTICIPANT SURVEY FORM – SOLUTION EQUATION SOURCE FLOWCHART

Code Basics Flow Chart Survey

	not used	strongly disagree	disagree	no opinion	agree	strongly agree
1	The basic instructions for using the Solution Equation Source (code basics) utility were clear and comprehensive					
	O	O	O	O	O	O
2	The definition given for a an INTERNALLY coded story problem was clear and concise (to the point)					
	O	O	O	O	O	O
3	The explanation and example(s) of INTERNALLY coded story problems helped me to understand this characterization					
	O	O	O	O	O	O
4	The definition given for a an EXTERNALLY coded story problem was clear and concise (to the point)					
	O	O	O	O	O	O
5	The explanation and example(s) of EXTERNALLY coded story problems helped me to understand this characterization					
	O	O	O	O	O	O
6	The Solution Equation Source (code basics) utility helped me to understand how the third of five characterizations in the categorization process is managed					
	O	O	O	O	O	O

APPENDIX G

PARTICIPANT SURVEY FORM – WORKED EXAMPLES UTILITY

Worked Examples Utility Survey

	not used	strongly disagree	disagree	no opinion	agree	strongly agree
1	The BASIC INSTRUCTIONS for using the Categorization Worked Examples utility were clear and comprehensive					
	○	○	○	○	○	○
2	In the automatic mode (as opposed to drag and drop) the INSTRUCTIONS for EACH STEP were clear and comprehensive					
	○	○	○	○	○	○
3	In the automatic mode (as opposed to drag and drop) the REASONS IN SUPPORT FOR EACH CHARACTERIZATION for each step were clear and comprehensive					
	○	○	○	○	○	○
4	In the automatic mode (as opposed to drag and drop) all computer behavior was normal and expected					
	○	○	○	○	○	○
5	In the drag and drop mode (as opposed to automatic) the INSTRUCTIONS for EACH STEP were clear and comprehensive					
	○	○	○	○	○	○
6	In the drag and drop mode (as opposed to automatic) the IMMEDIATE FEEDBACK (correct/incorrect) for EACH STEP was clear and comprehensive					
	○	○	○	○	○	○
7	In the drag and drop mode (as opposed to automatic) all computer behavior was normal and expected					
	○	○	○	○	○	○
8	I was able to switch between AUTOMATIC and DRAG AND DROP modes of operation while still maintaining the proper categorization sequence					
	○	○	○	○	○	○
9	The ability to switch between AUTOMATIC and DRAG AND DROP modes of operation helped me to understand the categorization process					
	○	○	○	○	○	○
10	The Worked Examples Categorization utility helped me to understand how the five characterizations determine the completed categorization process					
	○	○	○	○	○	○

APPENDIX H

GENERAL STUDY INFORMATION HANDOUT

General Study Information Handout before the Research Session Begins

General Information Sheet for Ms. Stephens' Categorization Practice Utility 1.0 (MS-CPU-1.0) Study

Thank you for participating in this study. The purpose of the study is to obtain information about the MS-CPU-1.0 with regards to its ease of use and its effectiveness in conveying the subject matter (story problem categorization) to you, the learner. Investigator – subject verbal interaction is encouraged through the research period, except for testing, as another avenue for information gathering. Once the information is collated and indicated changes have been completed, the new version will be subjected to a different set of students and teachers for their subsequent evaluations.

Both the student and teacher subject evaluations are conducted with the understanding that these research efforts are interactive with questions and comments during the actual use of the MS-CPU strongly encouraged from the standpoint of the subject enabling the investigator to accumulate as much information as possible regarding the improvement of the system. The study will consist of the following procedures in chronological order:

1. Read the General Information Booklet (presently reading)
2. Read the General Descriptions of the Categorization Process Booklet
3. Take a 12 question - story problem categorization pre-test
4. Use of the MS-CPU-1.0 to enable the collection of information about the system
    a. Use all aspects and sub-utilities available in the system to include
        i. Hyperlinked definitions
        ii. Hyperlinked graphical representations
        iii. Target method interactive flow-chart utility for determining problem quantity relations or lack thereof
        iv. Code basics interactive flow-chart utility for determining solution equation source
        v. Worked examples utility

vi. Interactive “calculator” for specialized terms and their associations with other terms

b. Investigator observation/note-taking while in use

c. A “think aloud” procedure where actions are audio-visually recorded

5. Take a 12 question - story problem categorization post-test
6. Take a survey/questionnaire about your use of the MS-CPU-1.0 tutoring system
7. An investigator/user follow-up interview
8. A free-response session (5 minutes) to wrap up your participation in the study

Individual information collected will be assigned a randomly chosen two digit number to disassociate the participant with their responses. Test results will be aggregated, and student/teacher observations will be listed in a table for each set of evaluation iterations associating the corrective observation with the number of times it occurred, a rationale for or against its remediation, and a description of the remediation as needed.

Thank you for your time and efforts to enable more GED students to pass the mathematics portion of the test…

ChristiAnne S Ritchey
MS-CPU Investigator

APPENDIX I

GENERAL CHARACTERIZATION DESCRIPTIONS HANDOUT

General Characterization Descriptions Handout before the Pre-test

General Descriptions Booklet for the Five Characterizations in the Categorization Process

Learning is the process of accumulating information about situations that are of interest to us. We might be interested in why a bee or a butterfly can fly even though they are apparently much different than other things that can fly, like a bird or a plane. The motivation for learning is usually generated from a curiosity about the thing or from a recognition that knowing about the thing can be useful in some capacity, such as for a test or as part of greater knowledge about something that is even more useful.

When information that is new is processed using short-term memory, it is obvious from its description that the information won't last or persist for very long. Unless we can commit the information to what is called long-term memory, the information won't be available to us at a later time or date, and we will have to repeat the same learning process as if it were new. This, of course, would be a waste of time, producing a situation where we will have to process it again and again. It would appear, then, that the path to meaningful learning is to simply make sure that what we are learning in the short term is committed to long-term memory; simple enough.

However, we would soon develop a labyrinth of inaccessible and unconnected information in long-term memory were it not for the ability of our brains to catalog new information according to a pertinent association with information that is already present in long-term memory. If I have a recipe for making my grandmother's delicious oatmeal cookies, my first attempt at making them will require close attention to the written recipe. However, as I *experience* making them, my thinking processes have automatically connected the various parts and methods of the recipe into a plan (called a **schema** in this study) that resides in long-term memory. The greater the quality and frequency of the experiences the quicker the **schema** will develop to a strength that no longer requires reading the recipe; understanding it and able to repeat it from **memory**.

Further, I may embellish the recipe for my purposes, and that information will be attached to the **schema** already held in long-term memory as well. Schema-building is only possible because I have cataloged or categorized my thing of interest as oatmeal cookies. If we are to learn truly, i.e., retain information, some method of categorization must be incorporated. Cookies are easy; the Revolutionary War is easy. Story problems, not as easy, but certainly doable.

The categorization process you are helping to research is the first step in Ms. Stephens' 5 step method for solving story problems and is named Ms. Stephens' Categorization Practice Utility (MS-CPU). The categorization process does NOT include a problem solution. However, it does serve two crucial purposes: As with the oatmeal cookies it establishes, then embellishes the information and processes that you incorporated during your experiences (both the successes and the failures) in solving a particular category or kind of story problem. The second purpose, using carefully chosen characterizations (5) to form a category is to familiarize you with the structure of the problem and how the remaining four steps will utilize the information found in the problem to solve it.

The story problem categorization process begins with the first characterization, **quantity number***.* Quantity Number asks you to determine whether the problem concerns a *single quantity* whose *value* is to be **changed** in some mathematical way OR the problem consists of **multiple** (two or more) quantities, either known (concrete, e.g. 5 dimes) or unknown (Bill has dimes in his pocket, which will be represented by a letter in a future step - 3). Detailed instructions are offered as you choose one of the two icons for this characterization and drag, and then drop it on the target.

The process continues with the second characterization which is to determine whether any **relationships** exist between the quantities previously itemized. This determination is quite easy for a change quantity problem because the new value of the single quantity will always have a mathematical relationship with the original value as that is how it is changed. Detailed

Instructions, including the Target Method, are offered to help you choose one of the three icons for this characterization and drag, then drop it on the target.

Next in the process is the third categorization which is to determine the **source of the solution equation** (*where* it is located). All solvable story problems have an associated equation(s) that, when solved, produce the answer to the question in the problem. If the source of the solution equation is entirely within the problem, which simply means that the equation can be synthesized or generated from the information contained in the problem, it is *internally coded*. If the solution equation is to be found outside the problem and pre-dates the story problem, it is an *externally coded* problem (e.g., the problem asks for the area of a rectangle). Here, code is synonymous with equation. Detailed instructions, including Code Basics Animation, are offered to help you choose one of the two icons for this characterization and drag, then drop it on the target.

Fourth in the process is showing the actual **solution equation**. However, this step is only completed when the problem has been characterized in the previous step as *externally coded*. Detailed instructions are offered as you choose one of several icons for this characterization and drag, then drop it on the target. If the problem is internally coded, the solution equation is generated in the 5th step of the story problem-solving process, and an auxiliary icon automatically drops into place when the internally coded icon is dropped in the third step.

The last step in the categorization process is to determine whether the **solution form** will be a concrete number (numerical) or as an expression (Y – 6). Detailed instructions are offered to help you choose one of the two icons for this characterization and drag, then drop it on the target.

This information serves to introduce you to the categorization process so that you have some idea of what you are supposed to be learning and some brief description of each characterization. This information is not expected to enable you to score high on the categorization pre-test, that is for the MS-CPU. It is here only to give a baseline measure so that the expertise you developed from using the MS-CPU can be measured. Once the pre-test has been completed, and you are within the "think aloud" protocol, questions for the researcher are accepted provided a reasonable effort has been made to acquire the information from the MS-CPU. Of course, questions will not be entertained when taking the post-test.

Please give your best efforts in using and evaluating this system as it is a new and innovative method for teaching story problem solution that has the potential to help many GED test-takers to pass the mathematics test and earn their GED diplomas.

Thank you!

APPENDIX J

MAIN CATEGORIZATION OBSTACLE IDENTIFICATION & MITIGATION-1ST ITERATION

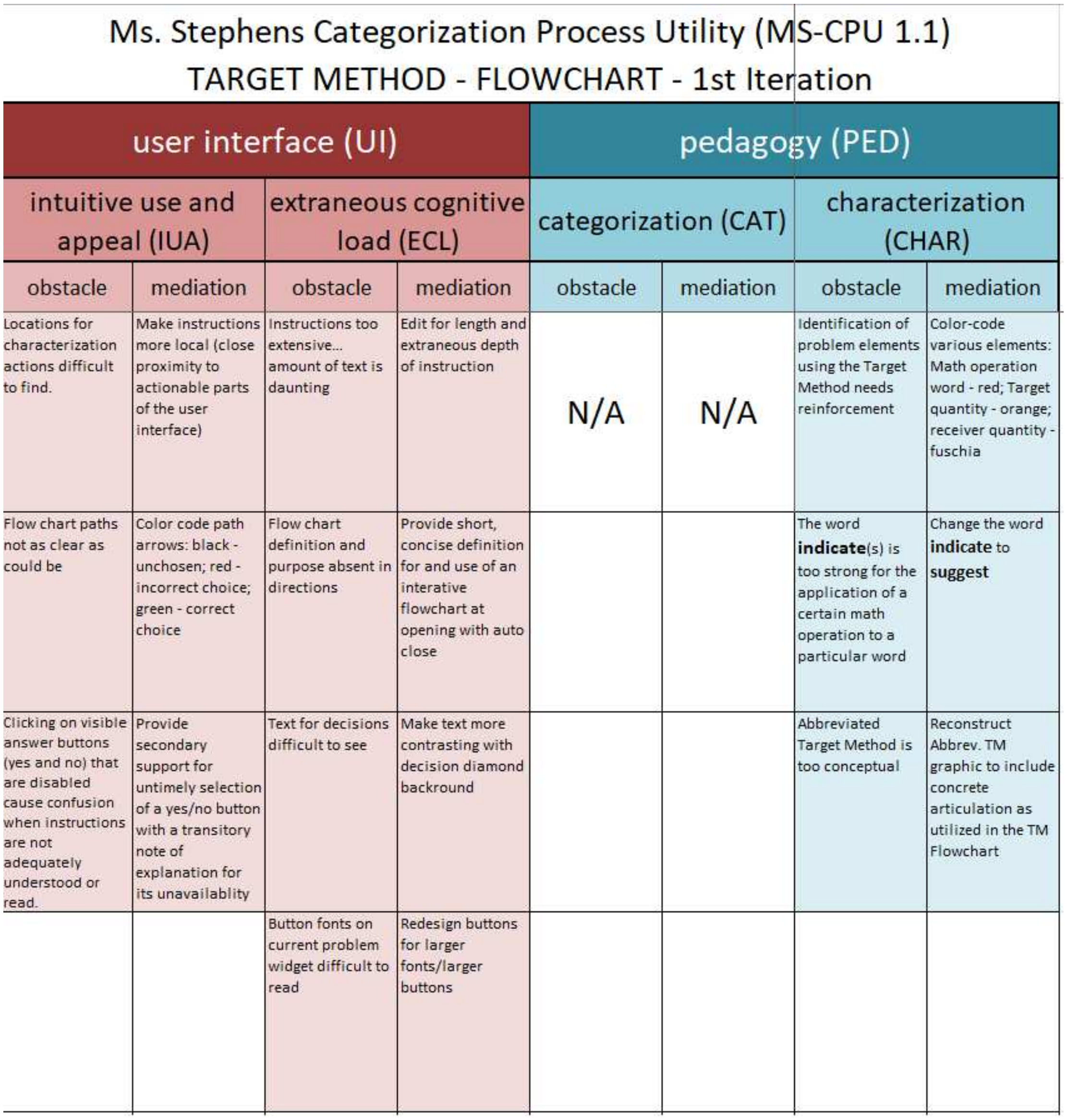
Ms. Stephens Categorization Process Utility (MS-CPU 1.1) MAIN - 1st Iteration

user interface (UI)				pedagogy (PED)			
intuitive use and appeal (IUA)		extraneous cognitive load (ECL)		categorization (CAT)		characterization (CHAR)	
obstacle	mediation	obstacle	mediation	obstacle	mediation	obstacle	mediation
Targets for drag/drop icons on categorization workspace not easily located. Not large enough/not a rectangle	Enlarge targets on categorization workspace and in the form of a rectangle	less than accurate identification of specialized terms in the problem text	Provide an option (button) to hilite problem quantity names that appear in the text	definition of the categorization process is missing	definition of problem categorization as a holistic approach that focuses on the interdependence of five characterizations.	The difference between a change quantity problem and a multiple quantity problem needs further elucidation	Distinguishing between a quantity that changes "over time" and two or more quantities that exist simultaneously or "co-exist" will help distinction clarity for determining the first characterization
May have missed clicking on a hyperlinked term or need the definition later, but is unavailable	Include a central dictionary of terms available from the Main Menu	"Why" reasoning text field contains not immediately recognizable terms	Add hyperlinks for terms that have specialized meaning or are not in common use	Importance of worked examples utility not clearly presented and easily available	After MSASPT - Categorization Step is presented make Worked Examples Utility automatically available whose rationale and instructions are required reading before Main program can be accessed	difficulty with definition of "multiple" quantity characterization to differentiate it from "change" quantity	emphasize "change" as taking place over time with problem containing a present or now situation and a past or future situation.
Blocking to prevent UI manipulation not obvious	Make blocking movie clip darker	Multiple Quantities draggable icon confusing as to whether it represents the number of quantities in a problem or not	Make multiple quantities icon reflect number of quantities in the problem (2-4)			difficulty with definition of "change" quantity characterization to differentiate it from "multiple" quantity	emphasize "multiple" as at least two quantities existing simultaneously or concurrently in the problem.
Hard to find specialized term on specialized terms widget	Alphabetize specialized terms on specialized terms widget	"according to Ms. Stephens' conventions" in problem text causes confusion	Remove this phrase from problem text that contains it			Internal/external distinction as purely whether or not a solution equation already exists which includes all named quantities in the problem is incomplete.	distinction must include named quantities that are presented in the form of a proportion, making the problem externally coded (proportion)
Unaware of instruction title and number	Enlarge font size to 20 of the step instruction in the instruction window and add accompanying and pertinent sound (voice).	"specialized terms" sub-utility not available on third step or fourth step characterizing instructions	hyperlink "specialized terms" sub-utility in 3rd and 4th step characterizing instructional text			too much information for "in terms of" phrase definition	minimize "in terms of" definition by illustrating Target Method for identifying relational pairs
		"Why" reasoning text field for providing reasons to the user for an error contains too much information and is daunting to the user	Edit "Why" textfields for brevity and effectiveness. Make font larger.			solution equation source definition needs clarification as "named quantities" are too nebulous.	added "with an assigned value - either known or unknown" to named quantities.

		math operations words sub-utility s/b available for second characterization on Main Menu	make math operations words available on the main menu			solution equation source definition needs clarification for "pre-existing" as it is too nebulous.	Give historical context to pre-existing with the derivation of the area of a circle (Archimedes, 250 bc)
		specialized terms w/associated terms sub-utility s/b available for third characterization on Main Menu	make specialized terms w/associated terms sub-utility available on the main menu				
		specialized terms w/associated terms sub-utility missing perimeter and average	create buttons and graphics for perimeter and average				
		Unaware of available help from hyperlinked terms	Write special note along with a clickable term (fuschia) to demonstrate the definition window.				
		Information and instruction are inter-woven and thus confusing	Separate information from characterization instruction. Put information first and "Do this…" manipulatives instruction second				
		less than accurate identification of named quantities in the problem text	Provide an option (button) to hilite problem quantity names that appear in the text				
		less than accurate identification of target method elements in the problem text	Provide an option (button) to hilite color-coded target method elements that appear in the text				
		less than accurate identification of math operation words in the problem text	Provide an option (button) to hilite math operation words that appear in the text				
		less than accurate identification of specialized terms in the problem text	Provide an option (button) to hilite problem special terms that appear in the text				

APPENDIX K

TARGET METHOD OBSTACLE IDENTIFICATION & MITIGATION-1ST ITERATION

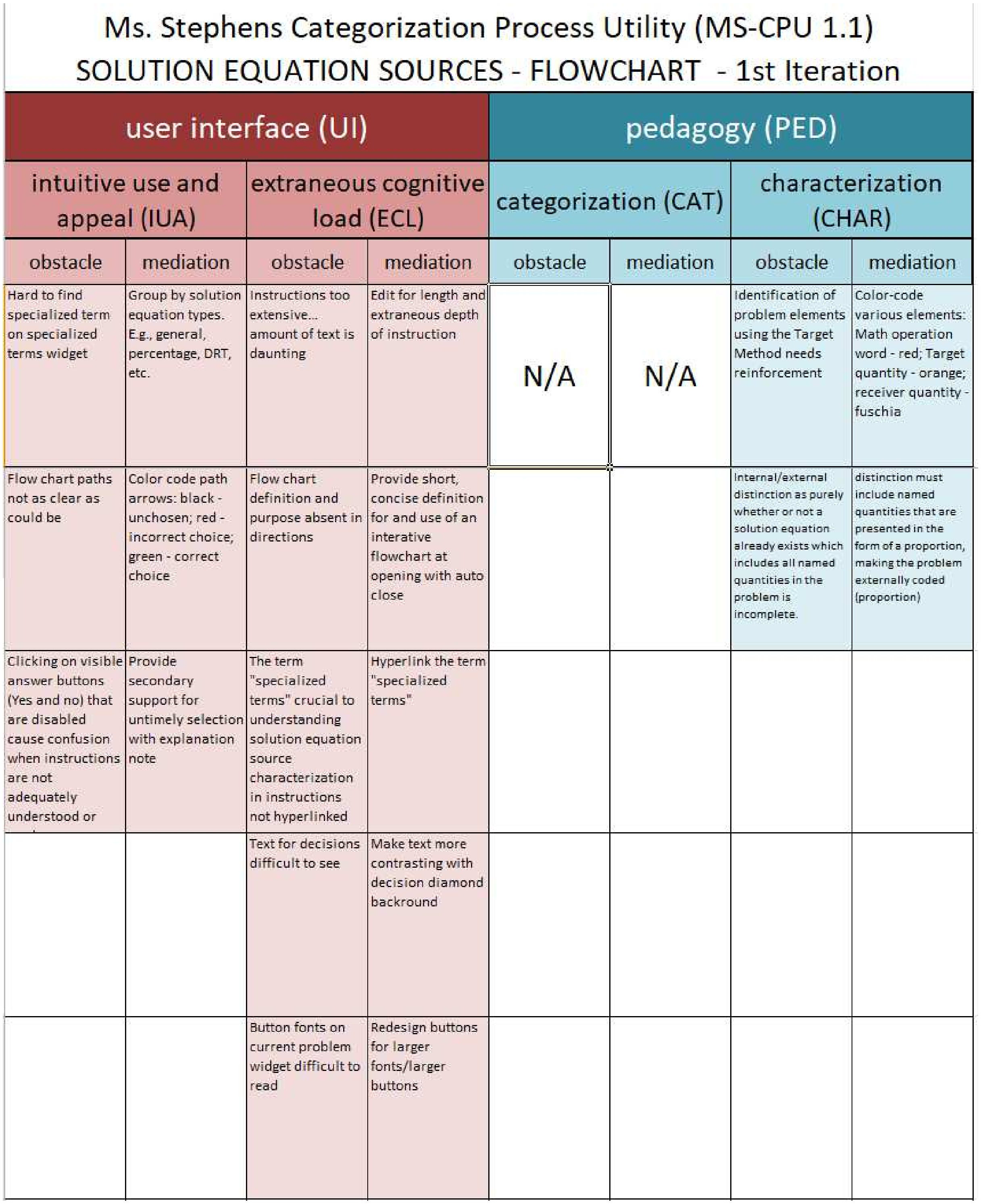


Ms. Stephens Categorization Process Utility (MS-CPU 1.1)
TARGET METHOD - FLOWCHART - 1st Iteration

user interface (UI)				pedagogy (PED)			
intuitive use and appeal (IUA)		extraneous cognitive load (ECL)		categorization (CAT)		characterization (CHAR)	
obstacle	mediation	obstacle	mediation	obstacle	mediation	obstacle	mediation
Locations for characterization actions difficult to find.	Make instructions more local (close proximity to actionable parts of the user interface)	Instructions too extensive... amount of text is daunting	Edit for length and extraneous depth of instruction	N/A	N/A	Identification of problem elements using the Target Method needs reinforcement	Color-code various elements: Math operation word - red; Target quantity - orange; receiver quantity - fuschia
Flow chart paths not as clear as could be	Color code path arrows: black - unchosen; red - incorrect choice; green - correct choice	Flow chart definition and purpose absent in directions	Provide short, concise definition for and use of an interative flowchart at opening with auto close			The word **indicate**(s) is too strong for the application of a certain math operation to a particular word	Change the word **indicate** to **suggest**
Clicking on visible answer buttons (yes and no) that are disabled cause confusion when instructions are not adequately understood or read.	Provide secondary support for untimely selection of a yes/no button with a transitory note of explanation for its unavailablity	Text for decisions difficult to see	Make text more contrasting with decision diamond backround			Abbreviated Target Method is too conceptual	Reconstruct Abbrev. TM graphic to include concrete articulation as utilized in the TM Flowchart
		Button fonts on current problem widget difficult to read	Redesign buttons for larger fonts/larger buttons				

APPENDIX L

SOLUTION EQUATION SOURCE OBSTACLE IDENTIFICATION & MITIGATION-1ST ITERATION



Ms. Stephens Categorization Process Utility (MS-CPU 1.1)
SOLUTION EQUATION SOURCES - FLOWCHART - 1st Iteration

user interface (UI)				pedagogy (PED)			
intuitive use and appeal (IUA)		extraneous cognitive load (ECL)		categorization (CAT)		characterization (CHAR)	
obstacle	mediation	obstacle	mediation	obstacle	mediation	obstacle	mediation
Hard to find specialized term on specialized terms widget	Group by solution equation types. E.g., general, percentage, DRT, etc.	Instructions too extensive... amount of text is daunting	Edit for length and extraneous depth of instruction	N/A	N/A	Identification of problem elements using the Target Method needs reinforcement	Color-code various elements: Math operation word - red; Target quantity - orange; receiver quantity - fuschia
Flow chart paths not as clear as could be	Color code path arrows: black - unchosen; red - incorrect choice; green - correct choice	Flow chart definition and purpose absent in directions	Provide short, concise definition for and use of an interative flowchart at opening with auto close			Internal/external distinction as purely whether or not a solution equation already exists which includes all named quantities in the problem is incomplete.	distinction must include named quantities that are presented in the form of a proportion, making the problem externally coded (proportion)
Clicking on visible answer buttons (Yes and no) that are disabled cause confusion when instructions are not adequately understood or	Provide secondary support for untimely selection with explanation note	The term "specialized terms" crucial to understanding solution equation source characterization in instructions not hyperlinked	Hyperlink the term "specialized terms"				
		Text for decisions difficult to see	Make text more contrasting with decision diamond backround				
		Button fonts on current problem widget difficult to read	Redesign buttons for larger fonts/larger buttons				

APPENDIX M

WORKED EXAMPLES OBSTACLE IDENTIFICATION & MITIGATION-1ST ITERATION

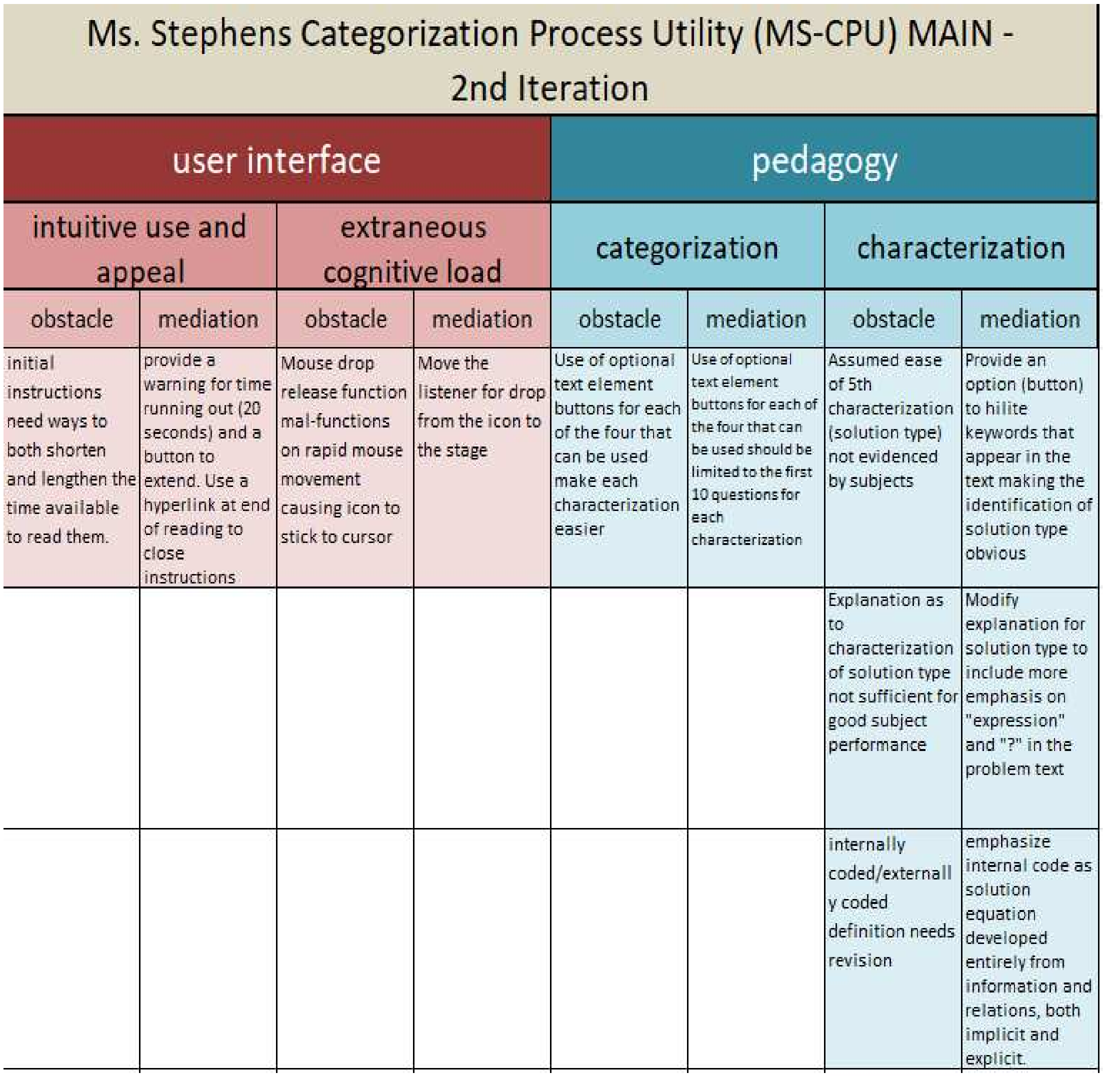
Ms. Stephens Categorization Practice Utility (MS-CPU 1.1) WORKED EXAMPLES UTILITY - 1st Iteration

user interface (UI)				pedagogy (PED)			
intuitive use and appeal (IUA)		extraneous cognitive load (ECL)		categorization (CAT)		characterization (CHAR)	
obstacle	mediation	obstacle	mediation	obstacle	mediation	obstacle	mediation
Targets for drag/drop icons on categorization workspace not easily located. Not large enough/not a rectangle	Enlarge targets on categorization workspace and in the form of a rectangle	Blue-bolded text mis-leading when not hyperlinked	Change live hyperlinks color to fuschia (#FF00FF) and underline	characterizations and their order are difficult to remember	add voice to each characterization and nominal separation in process	The difference between a change quantity problem and a multiple quantity problem needs further elucidation	Distinguishing between a quantity that changes "over time" and two or more quantities that exist simultaneously or "co-exist" will help distinction clarity for determining the first characterization
Next/Back button function not clear	Create mouse-over description of purpose for each button. Description to be used only once	Reset problem vs. Reset Utility not clear	Create mouse-over description of purpose for each button. Mouse-over to be cancelable by user			difficulty with distinction between "change" and "multiple" quantity characterization	emphasize "change" as taking place over time with problem containing a present or now situation and a past or future situation.
Next/Back button location not intuitive	Locate buttons at bottom of instructional / rationale text area	Information and instruction are inter-woven and thus confusing	Separate information from characterization instruction. Put information first			less than accurate identification of specialized terms in the problem text	Provide an option (button) to hilite problem quantity names that appear in the text
Hyperlinks not used - user unaware of utility	Incorporate an example in the general instructions that user must access before going to next step	Congratulations message too small	Make congratulations message larger			difficulty with distinction between "change" and "multiple" quantity characterization	emphasize "multiple" as at least two quantities existing simultaneously or concurrently in the problem.
sub-utility close buttons positioned on upper left leading to user not finding them	reposition sub-utility close buttons to lower right position	Characterization definition buttons not present	Install characterization buttons as done on the main Categorization utility			Internal/external distinction as purely whether or not a solution equation already exists which includes all named quantities in the problem is incomplete.	distinction must include named quantities that are presented in the form of a proportion, making the problem externally coded (proportion)
Example problem choice buttons difficult to locate	Set example problem choice buttons to blink in succession when needed fo user choice	TM and Code Basics Utilities missing	Make TM and Code Basics Utilities available				
User interface should be as much like main MS-CPU as possible	add TM Basics and Solution Equation Source Basics access buttons to workspace when in drag and drop mode at appropriate time.	problem text choice makes instructions, reasons, etc. difficult to read	place mask over problem text choice area after problem has been selected. Remove at problem completion				

Close instructions button needs to be present on all pages, not just first	Install close instructions button on all pages	Unaware of available help from hyperlinked terms	Write special note along with a clickable term (fuchsia) to demonstrate the definition window as part of the general instructions.				
Main Menu not easily located	Give more of a contrasting background color. Animated arrow pointing to its position at first-time use of utility.	back button function errors occur from time to time	convert back button function to restart the current step as opposed to going back one step				
Blocking to prevent UI manipulation not obvious	Make blocking movie clip darker	button icons on user instruction page are difficult to read	use mouse over to enlarge button icons (1.4) then mouse out to return to normal				
EXIT button not intuitively located	Locate in upper left and enlarge	less than accurate identification of named quantities in the problem text	Provide an option (button) to hilite problem quantity names that appear in the text				
reset problem and reset utility not intuitively located	make buttons larger and place in categorization workspace. Create mouse overs for increased visibility and purpose	less than accurate identification of target method elements in the problem text	Provide an option (button) to hilite color-coded target method elements that appear in the text				
general instructions for practice utility use not always available	make general instructions on Main Menu always available	less than accurate identification of math operation words in the problem text	Provide an option (button) to hilite math operation words that appear in the text				
		less than accurate identification of specialized terms in the problem text	Provide an option (button) to hilite problem quantity names that appear in the text				

APPENDIX N

MAIN OBSTACLE IDENTIFICATION & MITIGATION-2ND ITERATION

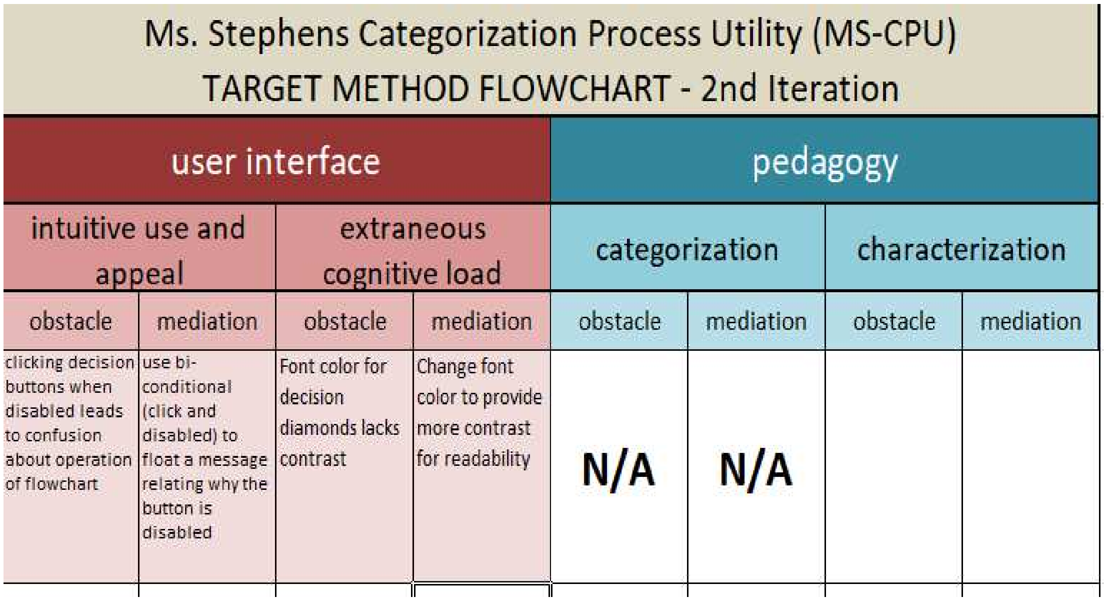


Ms. Stephens Categorization Process Utility (MS-CPU) MAIN - 2nd Iteration

user interface				pedagogy			
intuitive use and appeal		extraneous cognitive load		categorization		characterization	
obstacle	mediation	obstacle	mediation	obstacle	mediation	obstacle	mediation
initial instructions need ways to both shorten and lengthen the time available to read them.	provide a warning for time running out (20 seconds) and a button to extend. Use a hyperlink at end of reading to close instructions	Mouse drop release function mal-functions on rapid mouse movement causing icon to stick to cursor	Move the listener for drop from the icon to the stage	Use of optional text element buttons for each of the four that can be used make each characterization easier	Use of optional text element buttons for each of the four that can be used should be limited to the first 10 questions for each characterization	Assumed ease of 5th characterization (solution type) not evidenced by subjects	Provide an option (button) to hilite keywords that appear in the text making the identification of solution type obvious
						Explanation as to characterization of solution type not sufficient for good subject performance	Modify explanation for solution type to include more emphasis on "expression" and "?" in the problem text
						internally coded/externally coded definition needs revision	emphasize internal code as solution equation developed entirely from information and relations, both implicit and explicit.

APPENDIX O

TARGET METHOD OBSTACLE IDENTIFICATION & MITIGATION-2ND ITERATION

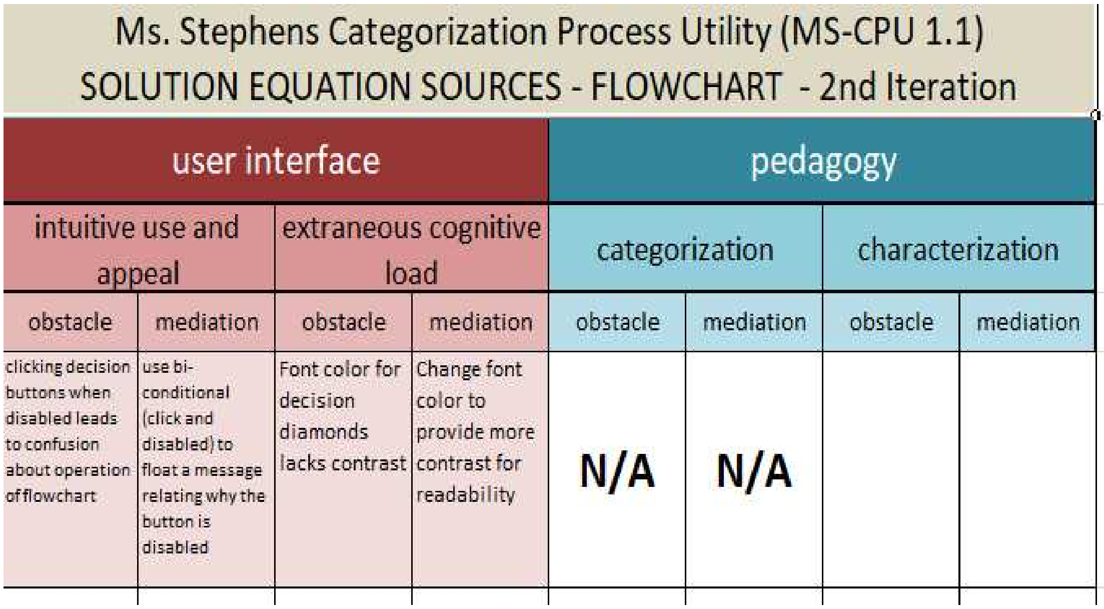


Ms. Stephens Categorization Process Utility (MS-CPU)
TARGET METHOD FLOWCHART - 2nd Iteration

user interface				pedagogy			
intuitive use and appeal		extraneous cognitive load		categorization		characterization	
obstacle	mediation	obstacle	mediation	obstacle	mediation	obstacle	mediation
clicking decision buttons when disabled leads to confusion about operation of flowchart	use bi-conditional (click and disabled) to float a message relating why the button is disabled	Font color for decision diamonds lacks contrast	Change font color to provide more contrast for readability	N/A	N/A		

APPENDIX P

SOLUTION EQUATION SOURCE OBSTACLE IDENTIFICATION & MITIGATION-2ND ITERATION

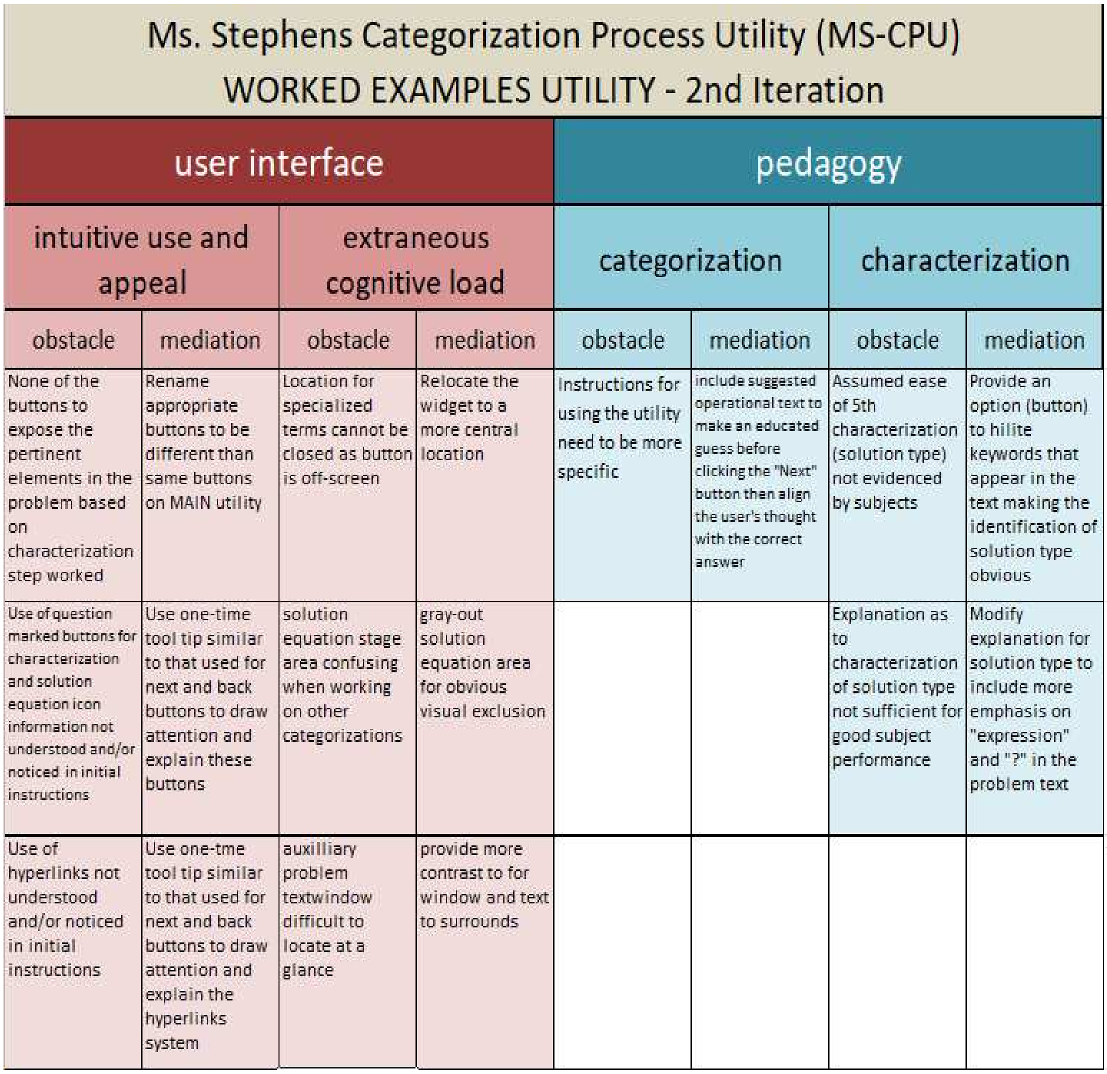


Ms. Stephens Categorization Process Utility (MS-CPU 1.1)
SOLUTION EQUATION SOURCES - FLOWCHART - 2nd Iteration

user interface				pedagogy			
intuitive use and appeal		extraneous cognitive load		categorization		characterization	
obstacle	mediation	obstacle	mediation	obstacle	mediation	obstacle	mediation
clicking decision buttons when disabled leads to confusion about operation of flowchart	use bi-conditional (click and disabled) to float a message relating why the button is disabled	Font color for decision diamonds lacks contrast	Change font color to provide more contrast for readability	N/A	N/A		

APPENDIX Q

WORKED EXAMPLES OBSTACLE IDENTIFICATION & MITIGATION-2ND ITERATION



Ms. Stephens Categorization Process Utility (MS-CPU)
WORKED EXAMPLES UTILITY - 2nd Iteration

user interface				pedagogy			
intuitive use and appeal		extraneous cognitive load		categorization		characterization	
obstacle	mediation	obstacle	mediation	obstacle	mediation	obstacle	mediation
None of the buttons to expose the pertinent elements in the problem based on characterization step worked	Rename appropriate buttons to be different than same buttons on MAIN utility	Location for specialized terms cannot be closed as button is off-screen	Relocate the widget to a more central location	Instructions for using the utility need to be more specific	include suggested operational text to make an educated guess before clicking the "Next" button then align the user's thought with the correct answer	Assumed ease of 5th characterization (solution type) not evidenced by subjects	Provide an option (button) to hilite keywords that appear in the text making the identification of solution type obvious
Use of question marked buttons for characterization and solution equation icon information not understood and/or noticed in initial instructions	Use one-time tool tip similar to that used for next and back buttons to draw attention and explain these buttons	solution equation stage area confusing when working on other categorizations	gray-out solution equation area for obvious visual exclusion			Explanation as to characterization of solution type not sufficient for good subject performance	Modify explanation for solution type to include more emphasis on "expression" and "?" in the problem text
Use of hyperlinks not understood and/or noticed in initial instructions	Use one-tme tool tip similar to that used for next and back buttons to draw attention and explain the hyperlinks system	auxilliary problem textwindow difficult to locate at a glance	provide more contrast to for window and text to surrounds				

APPENDIX R

LITERATURE REVIEW OF PROBLEM-SOLVING RESEARCH BEFORE 1950

An ability to solve problems has been a necessity, depending on your stance regarding creation, for those of the human persuasion for at least many millennia. Certainly, there is boundless evidence of earliest man solving problems with tools designed, then made by hand for a particular purpose; if Adam is your wont, he was solving huge problems right from the beginning. Actual descriptions of problem-solving methods probably began to be taught in ancient Greece and in the Mayan culture around 400 BC. In Greece, Aristotle was teaching his theory of means-ends-analysis around 340 BC, which can be found in the chapter entitled "The nature of deliberation and its objects" of his book, the *Nicomachean Ethics* (Book 111. 3, 1112b).

Aristotle's means-ends-analysis (MEA) heuristic:

We deliberate not about ends, but about means. […We] assume the end and consider how and by what means it is attained; and if it seems to be produced by several means they consider by which it is most easily and best produced, while if it is achieved by one only they consider how it will be achieved by this and by what means this will be achieved, till they come to the first cause, which in the order of discovery is last. . . and what is last in the order of analysis seems to be first in the order of becoming.

Aristotle also advanced elements of behaviorism in his books, *De Anima* and *De Motu Anima*, but his MEA efforts continue as a mainstay in many aspects of problem-solving and problem-solving research.

In more recent history, Rene' Descartes, in Discourse on Method: Part II, described the method he used for his problem-solving such that, "(…) instead of the great number of precepts of which Logic is composed, I believe that the four following would prove perfectly sufficient for me, provided I took the firm and unwavering resolution never in a single instance to fail in observing them."

1. Never to accept anything for true which I did not clearly know to be such; that is to say, carefully to avoid precipitancy and prejudice and to comprise nothing more in my judgment

than what was presented to my mind so clearly and distinctly as to exclude all ground of doubt;

2. Divide each of the difficulties under examination into as many parts as possible, and as might be necessary for its adequate solution;
3. To conduct my thoughts in such order that, by commencing with objects the simplest and easiest to know, I might ascend by little and little, and, as it were, step by step, to the knowledge of the more complex; assigning in thought a certain order even to those objects which in their own nature do not stand in a relation of antecedence and sequence;
4. In every case to make enumerations so complete, and reviews so general, that I might be assured that nothing was omitted;

Certainly, the second and third precepts above have a significant place in problem-solving efforts. He further states, "The long chains of simple and easy reasonings by means of which geometers are accustomed to reach the conclusions of their most difficult demonstrations, had led me to imagine that all things, to the knowledge of which man is competent, are mutually connected in the same way, and that there is nothing so far removed from us as to be beyond our reach, or so hidden that we cannot discover it, provided only we abstain from accepting the false for the true, and always preserve in our thoughts the order necessary for the deduction of one truth from another”. Rather more inspirational than operational, but it is at the heart of problem-solving as much as any other characteristic.

The efforts that these two seminal figures in world history gave to find methods for problem-solving clearly demonstrates the importance and the necessity for all of us to become reasonably adept at problem-solving in as many domains as concern us. In our particular place in history, that would include story problem-solving (the practical application of fundamental mathematics). Such has been described by many respected practitioners in the field of education (Gagne’, 1980; Schoenfeld, 1992; Lester, 1994; Jonassen, 2000) and by the National Council of Teachers of Mathematics (NCTM, 1980; 1989; 1991) over the last 70 - 80 years. To this purpose, several of the previously mentioned people, as well as many other educational theorists and psychologists, have produced ideas and methods that have advanced the field of problem-solving including story problem-solving. Having begun the examination of problem-solving methods with Aristotle's and Descartes' views to fix the chronology, the historical review of research and methodological efforts in the domain of problem-solving continues into the modern era.

The three seminal figures of Gestalt psychology, Max Wertheimer, Kurt Koffka and Wolfgang Köhler, conducted the first experimental work in problem-solving. Gestalt psychology (Gp) is the study of how people integrate and organize perceptual information (things seen, heard, felt, etc.) into meaningful wholes. Köhler characterized this study when he said, "the whole is other (or different) than the sum of its parts." The essence of Gestalt psychology is to emphasize the structural quality of the way in which we perceive, think about, and feel, the world around us. This structural quality is wholeness ('Gestalt' means 'whole' in German) (Laurillard, 1997). Five principles of perceptual organization are associated with Gp:

1. The law of *similarity* suggests that similar things tend to appear grouped together.
2. The law of *Pragnanz* holds that objects are seen in a way that makes them appear as simple as possible.
3. The law of *proximity* holds that things that are near each other seem to be grouped together.
4. The law of *continuity* holds that points that are connected by straight or curving lines are seen in a way that follows the smoothest path.
5. According to the law of *closure*, things are grouped together if they seem to complete some entity.

Karl Duncker, a Gestalt psychologist who studied under Koffka and Köhler, developed *functional fixedness* as a problem-solving ability determinate using the law of *Pragnanz* to construct what is called Duncker's Candle Problem*. The person to be tested was given a box of tacks, some matches, and a candle and instructed to fix, then light a candle on a wall in such a way that the candle wax doesn't drip onto the table below.* Once the person removed the tacks (*as an act of insight*) from the box, the solution proceeded efficiently. It did, however, take a good degree of creativity and insightful thought to get to that point as, according to the law of *Pragnanz,* most would perceive the materials in their simplest form, seeing only the whole instead of examining its constituent parts as they search for a solution. The ability to partition the whole, which opposes this law, would allow for more creative and insightful problem-solving. Taking the tacks *out of the box*, (pun intended) demonstrates a low level of functional fixedness and thus, a high level of problem-solving ability.

Another Gestaltist application to problem-solving came from Abraham Luchins in 1942. He postulated the *Einstellung effect*, which stated that a problem-solver will use a solution plan that is similar to something that worked on a comparable problem in their past even though a more effective solution path exists, one that will yield the correct answer. The avoidance of the *Einstellung effect* is consistent with Gestalt problem-solving theory that emphasizes creative thinking as well as insight (which may be sudden) in the solution of a problem.

In 1926, Graham Wallas offered a four-step approach to problem-solving that emphasized creative thinking and inspiration using Gestalt principles.

1) Preparation: defining the problem and gathering information relevant to its solution;
2) Incubation: thinking about the problem at a subconscious level while engaging in other activities;
3) Inspiration: having a sudden insight into the solution of the problem;
4) Verification: checking to be certain that the solution is correct;

Its general applicability precludes its consistent use in story problem-solving, but, even in my own experiences with solving problems, allowing for an incubation period most certainly facilitated the inspiration that enabled the solution of a problem.

George Pólya, the father of modern problem-solving (1954), proposed a general empirical process for solving problems that loosely resemble Descartes' steps and has been very influential in designing universal problem-solving pedagogy (Schoenfeld, 1992) and even in present-day cognitive science's investigations of human story problem-solving processes.

1) Understanding the problem: identifying the problem's knowns (givens) and unknowns and, if appropriate, representing the problem with the proper notation or symbols;
2) Devising a plan: determining appropriate actions to take to solve the problem;
3) Carrying out the plan: executing the actions that have been determined to solve the problem and checking their effectiveness;
4) Looking backward: evaluating the overall effectiveness of the approach to the problem, with the intention of learning something about how similar problems may be solved on future occasions. (instances of both meta-cognition and schema construction);

Wallas and Pólya generated their problem-solving methods from an analysis of their personal methods and informal observations through the lens of Gestalt psychology. Many of the mathematical intricacies used in more modern times to model the cognitive processes in problem-solving had not yet been introduced so that they could be scientifically investigated by enabling machine simulation of such skill (Wang, & Chiew, 2010). Even as these capabilities became available, almost all research on mathematical problem-solving traces back to Polya's insight (Teaching Undergraduates Mathematics, 2009).

During the 1920's, Gestalt Theory was widely advocated for the study of human cognition. Several Gestaltists were researching and writing during this decade: Kohler wrote. "The Mentality of Apes" in 1925 and Duncker produced "A qualitative study of productive thinking" in 1926. By the early 1930s, however, Gestalt psychology was subordinate to B. F. Skinner's *operant conditioning* version of behaviorism, a much more attractive (at the time) approach to the investigation of why humans behave the way they do, and a movement antithetical to Gestalt principles, for its emphasis on investigating and accepting only physically observable behavior as opposed to investigating internal events like thinking and emotion. From its inception with John B. Watson's, "Psychology from the Standpoint of a Behaviorist," and its continued development with the works of Pavlov (classical conditioning) and Skinner (operant conditioning), behaviorism accounted for a number of psychological phenomena. The behavioral explanations for these phenomena attracted many psychologists, especially younger ones, to the principles of behaviorism. Moreover, it is probable that the attraction resulted from an entirely different way to look at psychology: through the lens of what was thought at the time to be pure science.

Contributing as well to the popular support of behaviorism by the scientific community, and later, into other forms of the scientific study of learning, was B. F. Skinner's "Learning Machine." Such a machine established the feasibility of such things and led to the "learning machines" that cognitive scientists, psychologists, and computer scientists have produced in the last several decades, although Skinner's machine was quite primitive in comparison to the "teaching machines" of today. Before moving on to the review of cognitive science literature as it regards problem-solving, this paper would be incomplete if it did not present other very powerful and positive contributions that behaviorists have made to educational psychology and meaningful learning. Paul Saettler (1990) stated six areas of impact that behaviorism has had and continues to have, on education in America:

1) The establishment of learning objectives should be used and specified in quantifiable and terminal behaviors (behavioral objectives);
2) B. F. Skinner's design as mentioned above of a teaching machine that used reinforcement to increase learning (operant conditioning);
3) B. F. Skinner's programmed instruction, which outlined instruction based on the theories of learning at the time;
4) Individualized instructional approaches which defined what is now known as *self-paced* learning;
5) The use of technology through hardware/software that was based on drill and practice;

6) A systems approach to instruction that was rooted in military and business models and required continuous evaluations and modification.

Another important contribution, though not on Saettler's list, is the notion that behavior is shaped when a teacher responds to a student's successes rather than his failures (On Purpose Associates, 2003).

In 1932, Edward Tolman wrote, "Purposive Behavior in Animals and Men," and later, "Cognitive Maps in Rats and Men," in 1948, to report on the results of his investigation into the connections between behavior and learning. He was a traditional behaviorist (objective research on nonhuman species), but was not as uncompromising as some of his colleagues, as he included cognitive elements in his explanations of how learning takes place. Tolman believed that there was merit to the Gestaltist concept of "latent learning" which postulated that learning isn't necessarily tied to reinforcement, a significant break with behaviorist theory. His experiments in support of this view involved rats that were allowed to investigate a maze without restriction and were then able to navigate the maze, absent reinforcement, and find the food that had been subsequently placed in the maze. Tolman was also concerned with behavior that might be connected to internal states in the subject, which, I suggest, might be a precursor to the affective and conative domains found in human cognition.

However, in spite of Tolman's efforts, behaviorism still had many unexplained phenomena, not least of which was problem-solving and knowledge transfer. From its beginnings in 1913, these failings and the self-important over-reach of its original claims vis-a-vis "to write a psychology [that would] never use the terms consciousness, mental states, mind, content, introspectively verifiable, imagery and the like to define it, and never go back on [its] definition" (Watson, 1994), brought its influence to its lowest ebb as the early 1950's arrived. Noted human learning researcher, Henry L. Roedinger III, described the behaviorism period this way: "(…) due to Watson, Skinner and their ilk, the Dark Ages descended – the religious orthodoxy of Behaviorism blanketed the land and smothered creative thought about cognitive phenomena and other topics” (Roedinger, n/d). A rather harsh indictment of behaviorism, but it does offer some evidence that the next wave of learning theorists had grown tired of the absolute and exclusionary stances of behaviorists, and were looking in other directions to fill some of the voids in behaviorist theory with, I suggest, some amount of pleasure and a nod to Edward Tolman.

While behaviorism was on the ascent and during its peak, Gestalt psychologists were still writing concerning problem-solving, even as behaviorists were not. Norman Maier wrote *Reasoning in Humans*, 1930; Karl Duncker wrote *Zur Psychologie des Produktiven Denkens (*Psychology of Productive Thinking*)*, 1935; George Katona wrote *Organizing and Memorizing*, 1940; Abraham Luchins wrote *Mechanization in Problem Solving*, 1942; again, Duncker wrote *On Problem Solving*, 1945; and the founder of Gestalt psychology, Max Wertheimer wrote *Productive Thinking*, 1945.

These and other Gestaltist writings with regard to problem-solving concentrated on multi-step tasks where only a few of the steps to be taken were crucial and more difficult than the others. Such problems are called *insight* problems because the solution follows rapidly once the crucial (insightful) step(s) have been taken (VanLehn, 1988). Examples of such tasks are Dunker's Candle Problem (functional fixedness – described previously), Luchins water jar refill problems (Einstellung effect - 3 water jars, each holding a different amount of water; with the task of figuring out how to measure a specified amount of water using only these jars), and Maier's two string problem (functional fixedness – tie the ends of two strings hanging from the ceiling together, with the ends of the string hanging down, wider than a person can extend both arms and long enough that they can be tied together, with a number of random articles in the room, including a tool that can be used as a plumb bob – e. g., pliers).

Additionally, an important label for these kinds of problems is that they are *knowledge-lean* as they do not require any relevant specific knowledge because most of the information needed to solve the problem is contained in the initial problem statement or knowledge of the problem domain is mostly incidental to its solution or even absent (Eysenck & Keane, 2000;

VanLehn, 1988). Other examples of knowledge-lean problems would be the *Tower of Hanoi puzzle*, the *15-puzzle, checkers (but not chess),* and *Let's Make a Deal (television show).*

APPENDIX S

THE WORKED EXAMPLES PROBLEM-SOLVING PROCESS

‘The road to learning by precept is long, but by example short and effective.' (Marcus Annaeus Seneca, ca. 4 BC–AD 65). Worked examples have always been a method for teaching problem-solving, and have an even longer history of use than Seneca. More than likely, at least as far back as Cro-Magnon man, 25,000 years ago, somebody had to find the first constructive use for fire, figure a way to create and control it and teach it to the next generation or someone else. Someone had to fashion a spear out of flint and use it to hunt for food, and then teach both the spear-making and the hunting skills to the next generation or someone else. In our time, probably the most ubiquitous instances of worked examples are Kahn Academy on the internet and virtually every teacher who has ever taught mathematics or any of the sciences. While we have all experienced worked examples in the service of teaching problem-solving, often, it has not been nearly as effective as one would think it should be. Given its long history of use, WE has shown itself to be less than adequate in mathematics and science classroom performance.

But the very definition of worked examples (WE) has changed over time. Historically, in a typical classroom lecture setting, WE had the teacher standing at the chalkboard, working a problem (most likely algorithmic in nature) and moving at a speed that virtually ensured completion of the lesson at just the right time for homework to be assigned and dismissal of class. Allowing not for those students who attempt to keep up, but can't, causing them eventually to drop out, either literally or figuratively. In the last 30-40 years, though, through advances in cognitive science, computers and their ability to be intelligent agents, WE instruction has become more effective for not only the students but for teachers as well. Research over the last 20 years has shown that worked examples are of great help for knowledge acquisition, especially in well-structured domains like mathematics, physics, or computer programming (Atkinson, et al. 2000), and they are thought by some to be the key process in most story problem instruction (Jonassen, 2003). In fact, considerable research has shown that problem-solving performance improved more after studying as few as two worked examples than from solving several well-structured problems (Cooper & Sweller, 1987; Sweller & Cooper, 1985; Ward & Sweller, 1990).

The most successful instructional technique for teaching students how to control their mathematical problem-solving strategies (metacognition) is cognitive modeling of problem-solving in context, that is, having a competent problem solver describe her thinking process as she solves a real problem in an academic setting (Mayer, 2001). This technique must include both a description by an experienced (though preferably not an expert) problem solver of how the problems are solved as well as the thought processes that the problem-solver used in gaining the solution, i.e., the meta-cognitive aspects of the solution. Moreover, it has been shown that worked examples enable learners to construct useful problem schemas, particularly useful as the student begins their story problem-solving study. The ability to categorize problems with similar solutions and construct solutions to novel problems by analogy to the example is also enhanced by the use of WE (Sweller & Cooper, 1985).

The inherent effectiveness of worked examples can be increased by the recognition of certain learning principles developed from computational models and the incorporation of the methods used to affect them in the worked examples process:

- Instruction-based Teaching – Instruction should facilitate dialog between the teacher and the student(s), as the teacher audibly relates the thought processes that are being used for the solution choices that are being made;
- Structure-emphasizing Example Sets – Multiple worked-examples should be available for the student to observe either from an in-person teacher/tutor or within a computer-based teaching system;
- Partial Completion – Problems are presented, and a solution is started, however, at some point in the process, the student must synthesize the step to continue the demonstration;

- Fading – As the problem-solver gains expertise, the teacher/teaching system withholds more and more of the solution demonstration, until the student has completed a problem solution un-aided;
- Self-Explanation – The student mimics the teacher's presentation of the processes, choices, and rationales used to find a solution; (Hilbert, et al., 2008).

APPENDIX T

THE MSASPT DESIGN PROCESSES AND FEATURES

MSASPT is designed to account for these philosophical, epistemological and pedagogical requirements by using an explicitly presented six-step ordinal process, with sub-steps for each, except the final algebraic solution step. The following describes each step, then, for greater clarity, exposited (in italics) for the story problem below:

A triangle's base is given as twice its height. If the area of the triangle is 64 square inches, what must be its base, in inches?

I. Categorization – an *explicit*, as opposed to the quasi-implicit categorization of problem types evident in most, if not all, problem-solving methods, facilitates the establishment of an anchor for subsequent schema construction, which includes problem-solving processes for that problem type. Classifications that comprise a problem categorization are typically dichotomous and consist of (1) number of quantities, (2) quantity relationships (trichotomous), (3) solution equation source, (4) solution equation, and (5) solution type.

    a. As the categorization process is the focus of this paper, a detailed treatment of this step takes place in Section VII.

    b. Example problem categorization: (1) multiple quantity; (2) relational; (3) externally coded, (4) A = bh/2; (5) numerical answer.

II. Quantity identification – A search of the problem text to find the names of quantities germane to the problem solution. Such a list may include quantities not related to the solution, but they will be eliminated from consideration by completion of this step.

    a. If a relationship between at least one *pair* of quantities exists (evident in the categorization step), an ordered list that represents the relational structure is constructed from the:

        i. Independent quantity (Iq). An Iq is a quantity upon whose value all other quantities are determined. It is possible for a problem to contain two Iq's which would require an equation system to solve.

ii. Dependent quantity (Dq). A Dq is a quantity dependent on the Iq for its value.

iii. Indirect dependent quantity (Idq); Idq. An Idq is a quantity dependent on a Dq for its value. One Idq can be dependent on another Idq as shown in the first list example. Possible combinations in a four-quantity problem are:

1. Iq; Dq_1; Idq_1; Idq_2
2. Iq; Dq_1; Dq; Idq_1
3. Iq; Dq; Dq_1; Idq_1
4. Iq; Dq; Dq; Dq
5. Iq_1; Dq_1; Iq_2; Dq_2
6. Iq_1; Dq_1; Dq_2; Iq_2

iv. Quantities not part of any paired relationship are unordered (Uo) and usually listed last.

b. If quantities (Q_1) and (Q_2) are related by total (T) an ordered list is constructed as:

i. Q_1; Q_2; T.

c. If quantities have no relationships list is unordered.

d. Relational structure discovery process is concretized by the student with the use of the Target Method (TM).

e. Example problem quantities ordered list: (Iq) height, (Dq) base, (Uo) area

III. The symbolic representation of each quantity germane to the problem solution. Symbolic representation can be concretized by the student with the use of the Concrete Articulation (CA) module.

a. Example problem symbolic representation: height = y; base = 2y; A = area (64 sq. in.)

IV. The textual representation of the solution equation:

  a. If the problem is characterized the categorization step as *internally coded* (see Section VI)

    i. Solution equation is concretely identified as a phrase or sentence in the problem text and written in the appropriate section of the user interface.

  b. If the problem is characterized in the categorization step as *externally coded,* this step is skipped as the solution equation already exists.

  c. This step is skipped as the *example problem is externally coded*.

V. Algebraic representation of the solution equation

  a. If the problem is characterized in the categorization step as *internally coded* (see Section VI)

    i. The solution equation is constructed from its textual template identified in step IV, using the symbolic representations generated in step III

  b. If the problem is classified in the categorization step as *externally coded* (see Section VI)

    i. The solution equation is a pre-existing formula congruent with the quantities contained in the problem text.

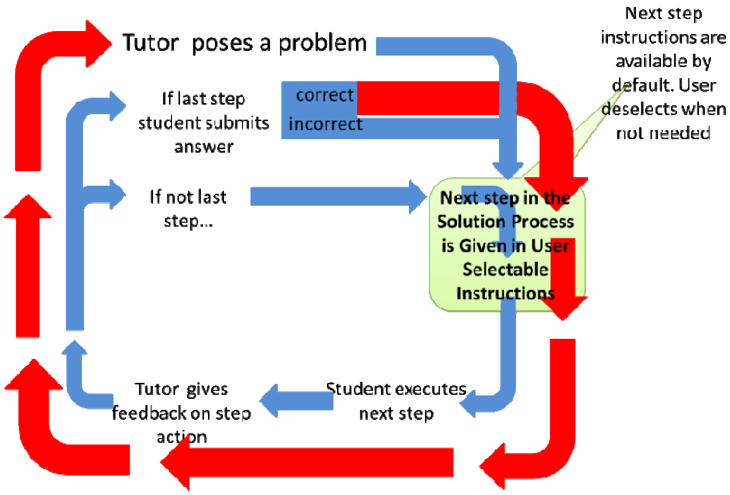
    ii. Example problem solution equation: $A = bh/2$.

  c.

VI. Numeric answer from the result of the solution equation

  a. Example problem numeric solution: $64 = (2y * y)/2$; $64 = 2y^2/2$; $64 = y^2$; $8 = y$ (height); since base = $2y$, triangle base must be 16 inches. Check: $16 * 8/2 = 64$.

MSASPT is a dual loop system, fulfilling the most basic requirement of an intelligent tutoring system, with its presentation of expertise via example-tracing (ETT). Appendix Q, Figure 1 below, illustrates the dynamics involved between the two loops in the MSASPT.


Next step instructions are available by default. User deselects when not needed
Tutor poses a problem
If last step student submits answer
correct
incorrect
If not last step...
Next step in the Solution Process is Given in User Selectable Instructions
Tutor gives feedback on step action
Student executes next step

Appendix Q, Figure 1. The only difference between the MSASPT and VanLehn's generic ITS is that the next step is explained in an instructions window contained in the user interface. Consequently, there is no need for next step hinting. Instruction access is voluntary after a certain level of performance is reached and has negative consequences if used beyond this point.

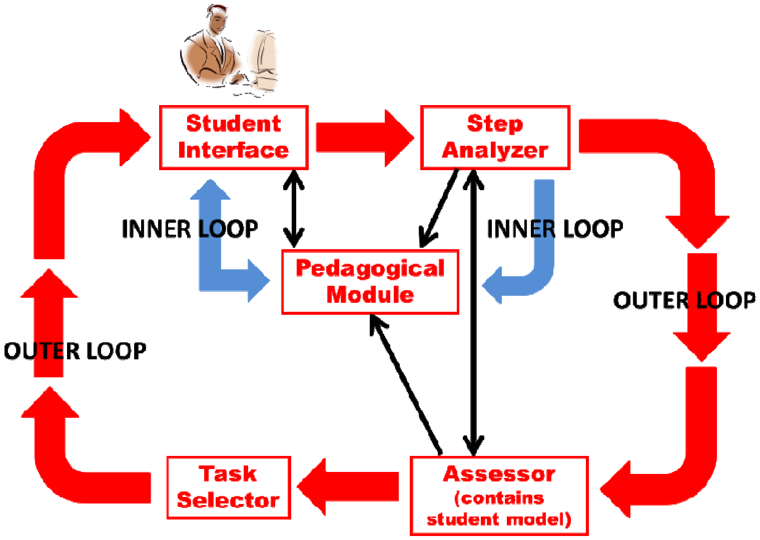
Student solution steps (and hint requests) are interpreted with respect to a multi-dimensional array whose primary and secondary indices contain the problem level and the problem number respectively for each problem contained in the tutor. The tertiary index contains relevant information according to the type of error and the level of the hint. Using arrays is structurally quite different from the directed acyclic graphs used in VanLehn's and other's ITS. The tutoring process consists of comparing student actions, which are constrained in the first four steps via drag and drop text or graphic tiles, against the hinting array for a specific problem. When a match for student input is found, a hint is placed in a separate array for access when the student elects to do so. Congruent responses allow the problem-solver to proceed to the next step. The use of arrays for recording possible problem-solving behaviors and their corresponding tutorial responses may well require more time in building the ITS, but directed, acyclic graphs proved to be beyond the ken of this computer programmer.

MSASPT: Outer loop – is quasi macro-adaptive as it uses a student model that consists of both step and sub-step performance metrics with respect to knowledge component acquisition to select each task at a particular level of difficulty. A skill level graph as part of the user interface is also updated as knowledge competency, and problem performance improve.

MSASPT: Inner loop - provides the five requisite services for each task as follows:

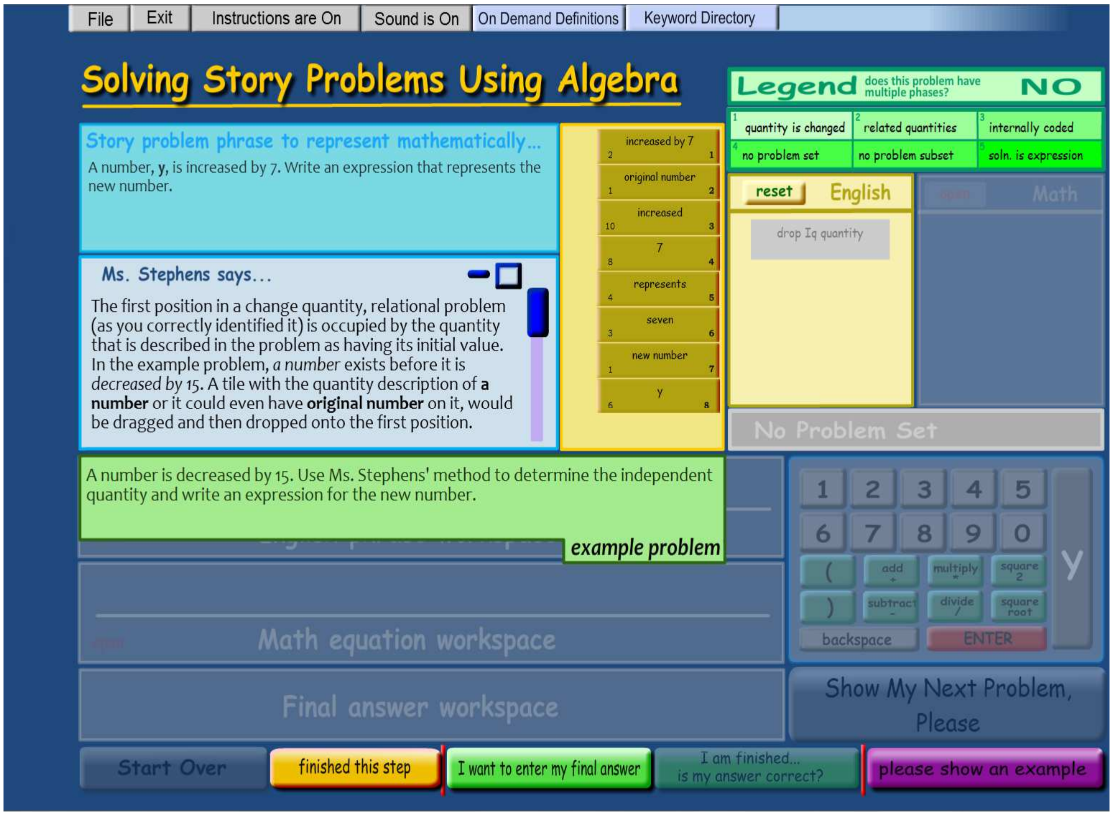
- MSASPT immediately flags two levels of responses: highlighted as correct (green), highlighted as incorrect (red).
- MSASPT only gives help (hints) when solicited. Help is available only on response errors as the steps in the solution process are available to the student by default.
    - Instructions can be turned off at some point in the learning process, by the student, which is requisite for advancement to the next level.
    - Instructions can be turned back on, at the expense of points needed for advancement.
- Hints are contextual and give advice on how the student may correct her/his response via the three levels (pointing, teaching, and bottom out) of hinting.
- MSASPT has a knowledge tracing module (student model) that monitors students' improvement of problem-solving skill based on both step and sub-step knowledge component acquisition. This information is used to determine the problem to be presented relative to difficulty and to advise the student when a skill seems to be consistently lacking.
- Upon completion of a varying interval of problems up to the total problems available, a comprehensive performance review is presented to the student along with any necessary recommendations, both remedial and congratulatory.

The MSASPT program architecture necessary to accomplish the various tasks attendant to the two loops in the system consists of five modules engineered to communicate with one another as shown in Appendix Q, Figure 2 below.



Appendix Q, Figure 2. An explanation of each module follows:

- *Student user interface* – the MSASPT UI is designed to clearly present the six-step problem-solving process with an intuitively designed problem space, a problem presentation window, an instructions window and a primary and secondary menu system. See Appendix Q, Figure 3 below for user the interface graphic.



Story problem phrase to represent mathematically...

A number, **y**, is increased by 7. Write an expression that represents the new number.

Ms. Stephens says...

The first position in a change quantity, relational problem (as you correctly identified it) is occupied by the quantity that is described in the problem as having its initial value. In the example problem, *a number* exists before it is *decreased by* 15. A tile with the quantity description of **a number** or it could even have **original number** on it, would be dragged and then dropped onto the first position.

A number is decreased by 15. Use Ms. Stephens' method to determine the independent quantity and write an expression for the new number.

example problem

- *Step Analyzer* – analyzes the student's performance to determine whether the input is correct or incorrect. The analysis would include other measures as well:
  - Correct knowledge components
  - Incorrect knowledge components that contain information as to specific remediation for a specific error.
  - Attribute-value pairs, where the attribute is the step skill and the value is the estimate that the student possesses that skill.
  - This information is passed on to both the pedagogical module and the student model.
- *Pedagogical Module* – The step analysis is stored in the pedagogical model to make feedback decisions about advice or hints given to the student, with additional input from the student model.

- *Assessor* - updates the student model that then becomes available to the pedagogical model.
- *Task Selector* - reads the student model after it has been updated by the analyzer and chooses the next task for the student. The chosen task is then sent to the practice environment.

One can correctly determine that MSASPT is similar, in many respects, to contemporary ITS's. However, looking below the surface, one can also see that it has been designed to be consistent in its features, services, and pedagogy, with the hypotheses that thinking in the abstract presents the most difficulty in learning how to solve story problems. Also, the incorporation of schema construction principles and schema theory in the ITS present the most practical path to problem-solving skill and its transfer to other types of problems in the domain.

The MSASPT Features:

Addressing other causes of story problem-solving difficulty from Section II, the MSASPT indirectly facilitates positive changes in the affective domain by approaching the difficulty from an entirely different direction, so that the problem-solving process can be knowable (understood) by both teacher and student. The tutor supports positive changes in the conative domain as incremental student success strengthens the desire (will) to complete the whole task successfully.

In regards to the lack of teacher expertise, the MSASPT was developed as much for the GED teacher as the GED student. Not only are most GED teacher's domains of expertise outside mathematics, but, for the most part, they suffer from the same difficulties in their story problem-solving efforts as the students do. The very same methods, processes and concretization opportunities that help the student develop effective story problem-solving routines are there for the instructors as well, thus mitigating the instructional inexpertise of many adult education teachers teaching mathematics.

Metacognitive factors such as being able to judge the difficulty of a task, to evaluate understanding of both text and process, to collect and use information towards reaching a goal or solution, and to estimate the ability to solve story problems all play a role in whether or not a student is on track towards story problem-solving acumen. For a tutoring system to both encourage and remediate metacognitive skills, it is important to draw a distinction between cognition and metacognition.

Cognitive abilities are used to achieve a particular goal (e.g., identifying the Iq) while metacognitive abilities are used to determine whether the goal has been reached (e.g., rationalizing the choice of the Iq) (Livingston, 2003). Metacognition usually precedes or follows a cognitive activity and occurs most often when cognition fails, such as finding out that the Iq selected was incorrect. Such mistaken notions are believed to activate metacognitive processes as the learner attempts to rectify the situation (Roberts & Erdos, 1993), and provide excellent opportunities for a tutoring system to address metacognitive issues. The MSASPT promotes metacognition through its instructional and hinting texts as well as in both single problem-solving summaries and its periodic problem-solving summaries.

As regards poor domain knowledge: in the MSASPT problem categorization process (detailed discussion in Section VII) a characteristic distinction is made between source coded and non-source coded story problems (Mayer, 1988). Briefly, according to Mayer, source coded problems have a solution code (equation) that must be sourced (located) from *outside* of the problem and is pre-existent and independent of the problem*.* An example problem: *If the length of a rectangle is twice its width and its perimeter is 60 inches, what is its length?*

The problem contains the quantities, length, width, and perimeter along with the descriptor, rectangle. Aside from discovering whether any relationships exist in the problem, the problem-solver need only match a pre-existing solution equation (from, say, a textbook or memory) with the quantities contained in the problem. This problem contains the quantities *length*, *width,* and *perimeter* so the solution code must be for the perimeter of a rectangle ($P = 2l + 2w$). In source coded problems, then, familiarity with certain domains (in this case, geometry) would be necessary. More on the depth of domain knowledge later.

For non-source coded problems the solution code (equation) is sourced from within the problem text (there exists no solution code from an external source, such as a textbook) An example problem:

Billy is three times as old as Azaria. In 10 years Billy will be twice as old as Azaria. How old is Billy now?

The solution code for this problem cannot be sourced from an exterior repository of possibly applicable codes (equations) and must be generated solely from the information in the problem text as,

- Billy is three times older than Azaria; let Azaria be **y** years old, then Billy must be **3y** years old.
- In ten years Azaria will be **y + 10,** and Billy will be **3y + 10**.
- In ten years Billy will be twice as old as Azaria.
- The solution code, written in English is Billy's age *equals* twice Azaria's age.
- The resultant solution code algebraically is **3y + 10 = 2(y + 10)**.
- Solving… **y(Azaria) = 10** and **3y(Billy) = 30**.

The GED Mathematics test taker must know where the solution code is located (source coded problems) as well as how to generate the solution code for non-source coded problems. Domain knowledge is required for source coded problems, but not to the extent of many other required demonstrations of knowledge (tests) as all test pertinent solution codes are available to the test taker in the test booklet. This testing context makes familiarity with each domain (geometry, trigonometry, uniform motion, etc.) a necessity, but not to the extent of many other testing venues, especially as found in physics domains such as motion, mechanics, nuclear, etc.

The MSASPT has on-demand information for the solution codes in the several domains that the GED Math test accesses and can be customized to add or delete domains according to the scope of problems to be solved. It is clear, though, that the designers of the GED Math test feel that domain knowledge can be retrieved in real-world problem-solving fairly easily, but that a familiarity with the included domains must be gained or the ability to retrieve such codes won't mean much.

A list of more explicitly presented design features along with an explanation of their operation in service of learning how to solve story problems follows:

- Hyperlinked term definitions and processes – terms that may be outside a common vernacular and graphical explanations of concepts and processes that appear in the instructions for a certain step, in error messages, or any form of communication between the student and the ITS are hyperlinked to a dictionary file for immediate presentation to the learner.
- Practice utilities for each of the story problem-solving steps – separate practice programs accessed from either the currently in-progress step or generally from a practice programs menu include an ITS for each of the five steps (categorization, quantity enumeration, symbolic representation, verbal equation representation, mathematical equation representation). All of the practice program ITSs include a worked example utility that can run each process automatically or be click-stepped through as well as toggling between the two modes.
- Worked examples for the problem solving process – a worked examples utility for the entire story problem-solving process that can run automatically or be click-stepped through as well as toggling between the two modes.
- Dictionary of terms and processes – an extensive dictionary of the terms, processes, and concepts that the student may encounter during their MSASPT experience. The dictionary is constructed as a tabbed folder system whose terms, processes, and concepts are keyed by their first letter.
- Target Method (TM); concretized instruction to determine relational structure – on-demand graphical explanation of TM principles and application to (1) determine whether a problem should be characterized in a certain way (detailed in Section VII) and (2) determine the

relational structure of the story problem as detailed above in the *processes* portion of this section. Essentially, the method accomplishes its purpose by:

1 Is there at least one instance of a mathematical operation in the problem text? (*e.g., Philip can run a mile in 15 seconds less than Raul*)

- Not found – the problem is *non-relational*
- Found (15 seconds *less*) – proceed to 2

2 Does the operation have a target for its action?

- No – the problem is *non-relational*
- Yes – proceed to 3 (less acts on Raul; there is a target [independent quantity] for the op.)

3 Does the result of the operation have a receiver?

- No – the problem is *non-relational*
- Yes – the problem is *relational* (Phillip receives the value of *Raul less 15* [dependent])

4 The use of the word "total" in the problem text must be differentiated between its use as a verb (a mathematical operation) and its use as a noun (the total of some number of quantities) -

- Verb – go to step 1
- Noun – go to step 5

5 Is there is a total value given for at least two quantities as a concrete number (numerical)?

- No – the problem is *non-relational* (i.e., the total is given as an unknown - symbolically)
- Yes – the problem is related-by-total

❖ Concrete Articulation (CA), concretized instruction to facilitate symbolic representation of problem quantities. Essentially, CA accomplishes its purpose by:

1. Assigning concrete values to the two quantities (relationships are always paired) in a relationship to determine how the dependent quantity's value changes as a result of the application of the mathematical operation.
2. Substitution of the symbol for the variable numerical quantity according to the concrete simulation.
3. For example: If a train travels twice as fast as a car…
    - The TM determines that the dependent quantity (receiver) is the train and the independent quantity (target) is the car.
    - Substitution of concrete quantities: car = 50 mph, so the train = 100 mph
    - What is the mathematical operation used: 2 times
    - If the car is **y,** then symbolization for the train is *two · y* or **2y**

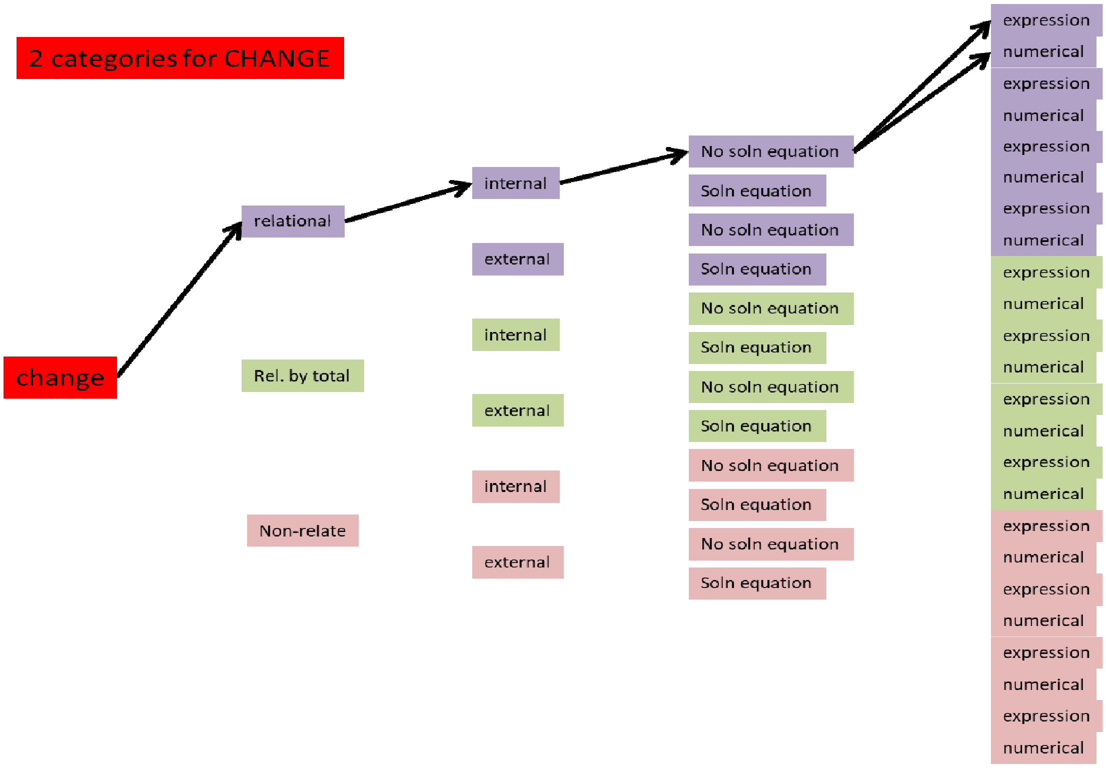
- ❖ On-demand flow chart of the problem-solving process – animated flow chart that requires the user to make the problem-solving decisions that s/he will make in an authentic story problem-solving experience
- ❖ Performance Report at varying problem intervals until competency is reached or problem bank is exhausted – data is imported from the student model and represents step performance as well as knowledge component acquisition.
- ❖ Progress Monitor – indicates by color (green, yellow, red) performance on each problem as well as when a student is ready for testing.

This concludes a description of the design and features of the MSASPT and how they work together to respond to the hypothesized single most important story problem-solving difficulty; an inability to think conceptually and make generalizations or abstractions.

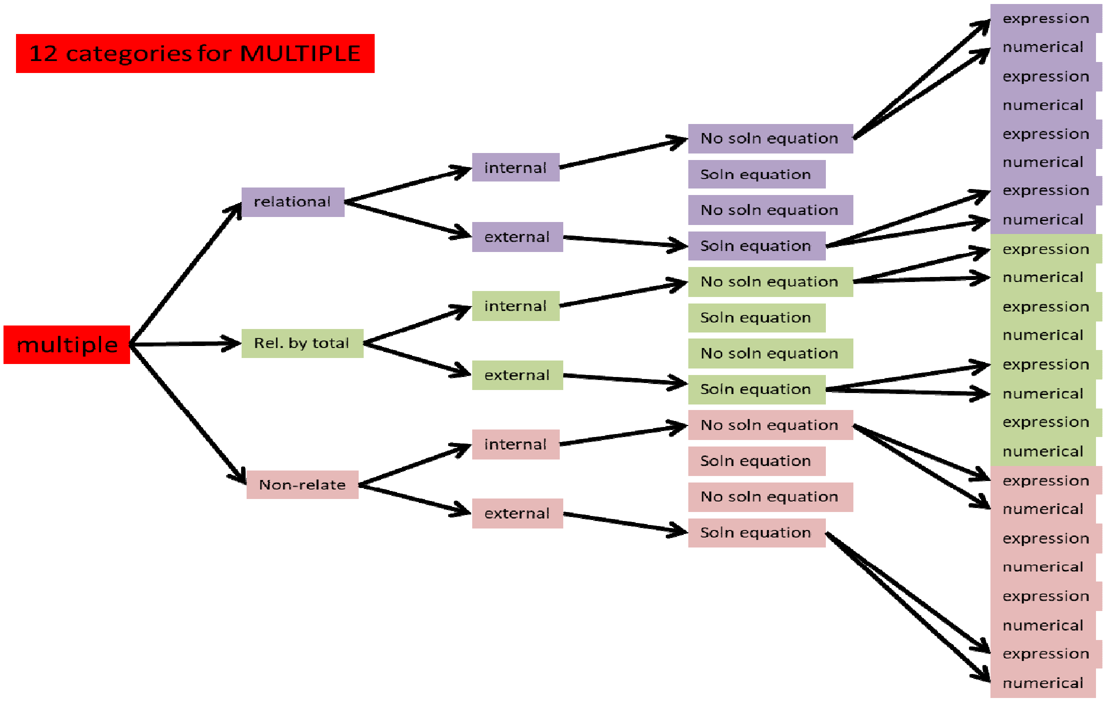
APPENDIX U

GRAPHIC REPRESENTATION OF CHARACTERIZATION COMBINATORIAL

2 categories for CHANGE

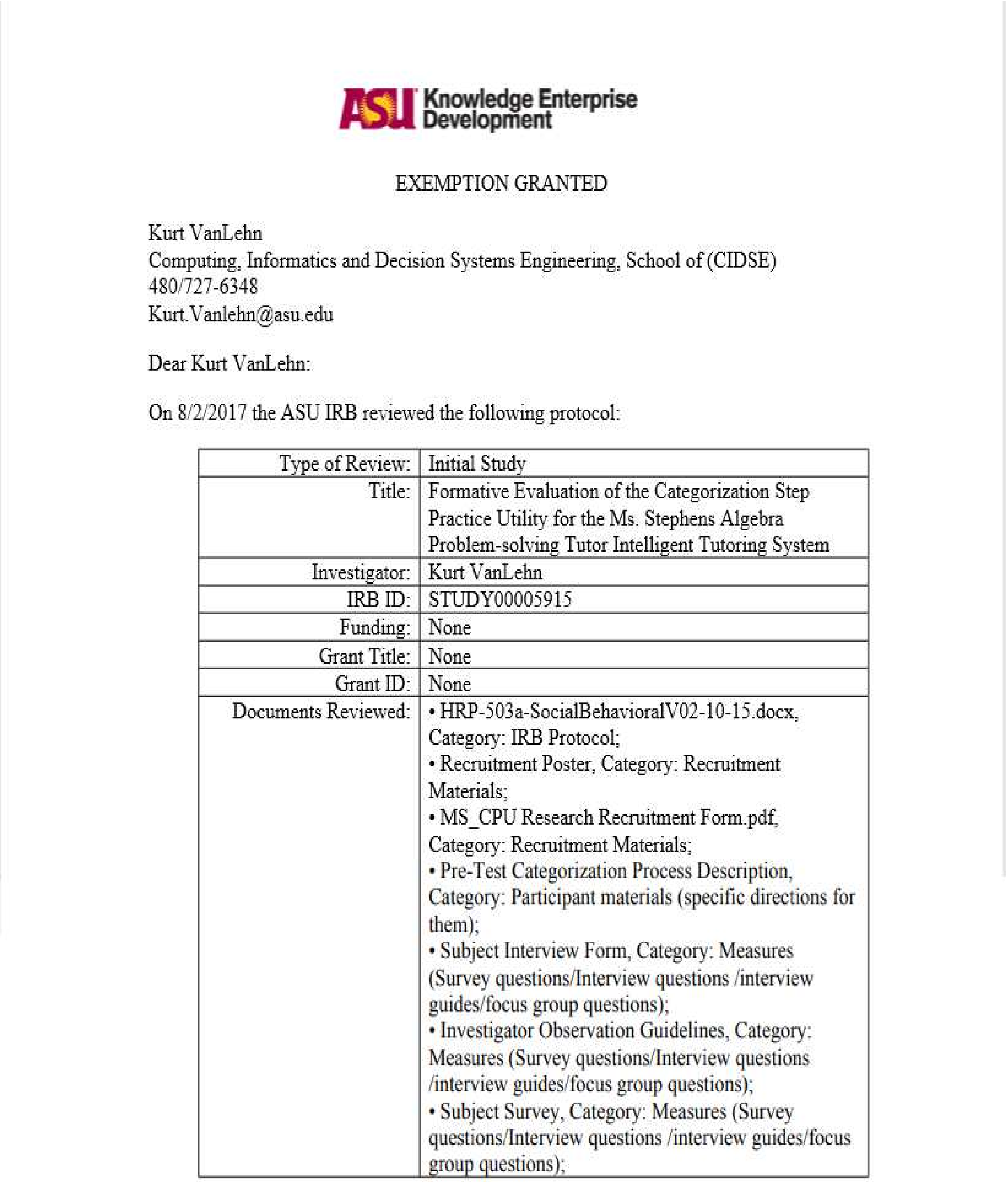


12 categories for MULTIPLE



APPENDIX V

IRB RESEARCH APPROVAL DOCUMENT



ASU Knowledge Enterprise Development

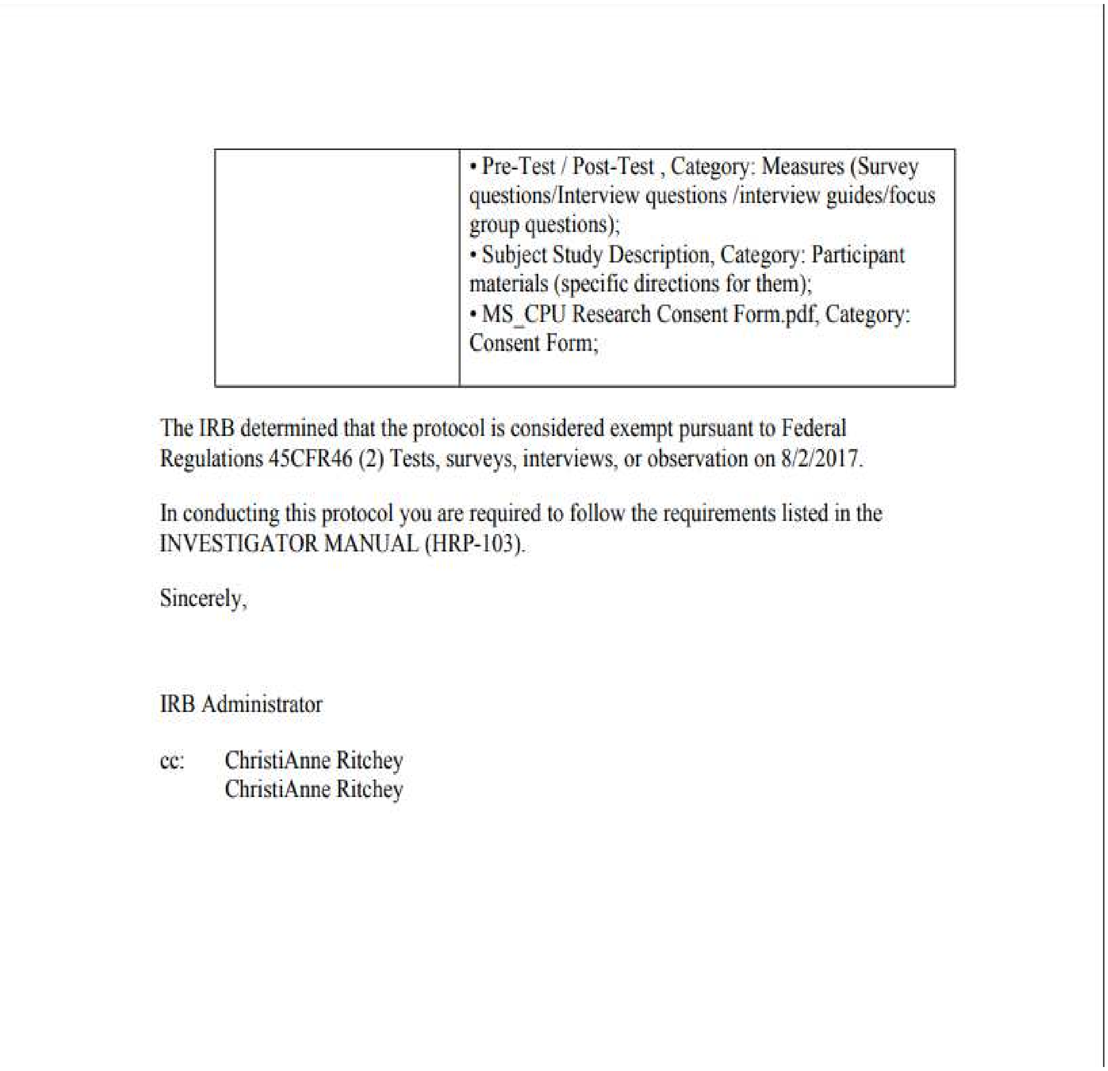
EXEMPTION GRANTED

Kurt VanLehn
Computing, Informatics and Decision Systems Engineering, School of (CIDSE)
480/727-6348
Kurt.Vanlehn@asu.edu

Dear Kurt VanLehn:

On 8/2/2017 the ASU IRB reviewed the following protocol:

Type of Review:	Initial Study
Title:	Formative Evaluation of the Categorization Step Practice Utility for the Ms. Stephens Algebra Problem-solving Tutor Intelligent Tutoring System
Investigator:	Kurt VanLehn
IRB ID:	STUDY00005915
Funding:	None
Grant Title:	None
Grant ID:	None
Documents Reviewed:	• HRP-503a-SocialBehavioralV02-10-15.docx, Category: IRB Protocol; • Recruitment Poster, Category: Recruitment Materials; • MS_CPU Research Recruitment Form.pdf, Category: Recruitment Materials; • Pre-Test Categorization Process Description, Category: Participant materials (specific directions for them); • Subject Interview Form, Category: Measures (Survey questions/Interview questions /interview guides/focus group questions); • Investigator Observation Guidelines, Category: Measures (Survey questions/Interview questions /interview guides/focus group questions); • Subject Survey, Category: Measures (Survey questions/Interview questions /interview guides/focus group questions);



	• Pre-Test / Post-Test , Category: Measures (Survey questions/Interview questions /interview guides/focus group questions); • Subject Study Description, Category: Participant materials (specific directions for them); • MS_CPU Research Consent Form.pdf, Category: Consent Form;

The IRB determined that the protocol is considered exempt pursuant to Federal Regulations 45CFR46 (2) Tests, surveys, interviews, or observation on 8/2/2017.

In conducting this protocol you are required to follow the requirements listed in the INVESTIGATOR MANUAL (HRP-103).

Sincerely,

IRB Administrator

cc: ChristiAnne Ritchey
ChristiAnne Ritchey