Student Errors in Dynamic Mathematical Environments

Molly Brown
West Lincoln High School; Lincolnton, NC;
(704) 736-9453 ext. 28131; Molly.Brown@lcsnc.org

Michael J. Bossé
Department of Mathematical Sciences, Appalachian State University; Boone, NC;
(828) 262-2862; bossemj@appstate.edu

Kayla Chandler
Department of Science, Technology, Engineering, and Mathematics Education; North Carolina State University; Raleigh, NC; kcchand2@ncsu.edu

Abstract

This study investigates the nature of student errors in the context of problem solving and Dynamic Math Environments. This led to the development of the Problem Solving Action Identification Framework; this framework captures and defines all activities and errors associated with problem solving in a dynamic math environment. Found are three different types of errors: domain (mathematics, technology, and problem solving), process (interpretation, activity, and evaluation), and interaction (syntactic and semantic). Implications from this study address problem solving in general.

Student Errors in Dynamic Mathematical Environments

Boolean operators take many mathematical forms including the conjunction (AND, $\wedge$) and the disjunction (OR, $\vee$) and many more. Most Dynamic Mathematical Environments (DME), such as The Geometer's Sketchpad (GSP), take advantage of Boolean operators in more advanced constructions. GSP users can indicate an action to occur when parameters or measures are equal to, less than or greater than others. The interconnectedness of the Boolean operators and GSP allow students to look at mathematics and the DME in novel and interconnected ways.

Mathematical representations (verbal, oral or written; numeric or tabular; symbolic or algebraic; and graphical), are central to student mathematical learning. As teachers observe how students interpret, generate, use, and translate between mathematical representations, this work illuminates student understanding of the underlying mathematics (Hollebrands, 2003). Therefore, the examination of student understanding through representations and their work in translating between representations is crucial for assessment of student mathematical understanding.

A translation is the mapping of one representation to another, for instance, creating a graph from a table of function values. Translating from one static representation to another has been studied for decades (e.g., Bossé, Adu-Gyamfi, & Chandler, 2014; Ainsworth & Van Labeke, 2004; Clement, Lockhead & Monk, 1981; Duval, 1999; Janvier, 1987; Kaput, 1987a, 1989; Knuth, 2000). However, with the novel introduction of DMEs, translations become more fluid and take on a less apparent nature. This leads to a renewed need to investigate the nature of students' work in respect to translations of mathematical representations in a DME.

This study, therefore, investigates the interplay of mathematical representations, transformations, and translations between representations within a DME and the nature of student understanding and misunderstandings. More specifically, this study examines types of errors that are made by students when attempting to solve an open-ended, Boolean activity in GSP.

Background Literature

This study considers the intersection of numerous fields previously investigated, some for decades, and pushes this nexus in novel directions. Some of these previous fields of study include: representations, transformation, and translations; dynamic mathematical environments and problem solving; Technological Pedagogical Content Knowledge (TPACK, formerly known as TPCK); and student error types when working with mathematical representations. Each of these dimensions is addressed in following sections.

Representations, Transformations, and Translations.

Mathematical representations are objects such as tables, graphs, words or symbols and are used to denote conceptual relationships (Cobb, Yackel, & Wood, 1992). Representations are used for communicating mathematical understanding, describing relationships between certain mathematical concepts, and operating with mathematical objects (Zhang, 1997). Connected with any representation are the conventions accepted for working with the representation.

Representations can be manipulated, or transformed, in order to better ascertain particular characteristics of the mathematical concepts contained in the representation. Transformations rewrite one form of a representation into another form of the same representation. For instance, a transformation on $x^2 + 5x + 6$ would be to factor the expression to obtain $(x + 2)(x + 3)$;

these expressions are different forms of the symbolic representation. Performing transformations can help students understand conceptual aspects encoded within representations (Edwards, 2003). For instance, while $x^2 + 5x + 6$ can be clearly recognized as a quadratic polynomial, concave up, with a *y*-intercept of 6, $(x + 2)(x + 3)$ can be seen as having two linear factors and two real roots, -2 and -3.

A translation occurs when the concepts contained in one mathematical representation (source representation) is mapped onto a different representation (target representation) (Bossé Adu-Gyamfi, & Chandler, 2014; Janvier, 1987), for instance, when a symbolic equation is converted into a graph. Figure 1 depicts the distinction between transformations and translations as the source representation can be transformed into another form of the same representation or translated into a different target representation. Rather than beginning from one representation and transforming it to another form or translating it to another representation (e.g., table and graph), the term *multiple representations* often denotes the simultaneous investigation of mathematical concepts through different and complementary representations (NCTM, 2000). Evidence shows that objects or concepts can be better understood when observed through multiple lenses (Edwards, 2003; NCTM, 2000).

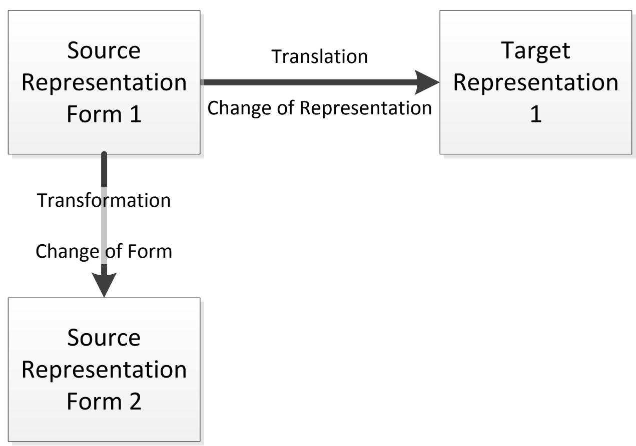

Figure 1. Transformations versus Translations.

Students use both representations and translations in all areas of mathematics. However, research shows that students tend to struggle with translations among mathematical representations (Garofalo, Drier, Harper, Timmerman, & Shockey, 2000) and more between some representational pairs than others (Bossé, Adu-Gyamfi, & Cheetham, 2011a, 2011b). It has been found that students may work within a representation and not actually understand the mathematical concept we want them to gain (Adu-Gyamfi & Bossé, 2014).

Particular processes have been recognized as students interact with mathematical representations. Some have recognized that the entire translation process can be encapsulated by three distinct yet interrelated activities: interpretation, implementation, and preservation (Adu-Gyamfi, & Bossé, 2014; Adu-Gyamfi, Bossé, & Chandler, 2015; Adu-Gyamfi, Stiff, & Bossé, 2012). These three activities can be restated as: interpretation, or the activity of interpreting either the source or target representation; action, or the activity of translating the source to target translation; and assessment, the activity of determining conceptual equivalence between the source and target representations.

Since it is often difficult to establish a one-to-one relationship between representations, interpretation of facts within each representation and across representations resolve ambiguities. For example, in order to graph a set of ordered pairs, one must interpret every ordered pair and associate each to a point on a coordinate plane. This is recognized as a *local interpretation* (Leinhardt, Zaslavsky, & Stein, 1990). However, in order to symbolically represent the relationship defined by a given set of ordered pairs, one must recognize how the coordinates of each ordered pair are changing in relationship to each other and the overall variation occurring among the ordered pairs of the set. This is recognized as a *global interpretation* (Duval, 2006). Students have more difficulty and perform more errors in respect to translations requiring global as opposed to local interpretive actions (Bossé et al, 2011a, 2011b; Adu-Gyamfi et al., 2012; Dreyfus & Eisenberg, 1987; Dunham & Osborne, 1991; Gagatsis & Shiakalli, 2004).

In respect to interacting with mathematical representations, Kaput (1987a, 1987b) recognized that some students employed semantic elaboration while other students employed syntactic elaboration. Syntactic elaboration connotes *locally* considering attributes and characteristics of a representation devoid of *globally* understanding the mathematical meaning encoded in the representation. Semantic elaboration can be viewed as *globally* understanding the conceptual information encoded in the representation without undue concern for precise *local* characteristics and attributes. For instance, the syntactic perspective may cause a student to look at $x^2 + 2x + 3$ as simply a concatenation of mathematical symbols whereas semantic perspective would allow the understanding of this expression as a quadratic polynomial with complex roots.

A student who can perceive $y = x^2 + 2x + 3$ as a quadratic equation and understands the correlated nature of a parabolic graph is translating between representations. Some would consider this student a mathematical thinker, or one who takes information from one representation and can easily create another representation that precisely encapsulates the same ideas (DePeau & Kalder, 2010). Thus, mathematical thinking includes semantic elaboration in the interpretation of the source representation, in the action of performing the representational translation, and in the assessment of equivalence between the source and target representations.

Domains of Knowledge

Technological Pedagogical Content Knowledge (TPACK, formerly known as TPCK) is a framework through which researchers can assess teacher knowledge and skills in respect to the interconnections of technological, pedagogical, and content knowledge (Koehler & Mishra, 2005; Mishra & Koehler, 2006; Niess, 2005, 2006). The TPACK framework provides an analytical lens with which to look at the instructional decisions that teachers make (Graham, Borup, & Smith, 2012) and can be applied to help educators effectively teach content in a technological environment (Schmidt, Baran, & Thompson, 2009). While this framework has significant value for teachers using technology (Zelkowski, Gleason, Cox, & Bismark, 2013), the participants of this study are not teachers and would be using the relevant technology to problem solve rather than make pedagogical decisions.

Borrowing the notion that research can simultaneously consider multiple domains of knowledge from the TPACK framework, this study develops a lens through which to observe and analyze student work in a DME in the domains of technology, content, and problem solving. Technological knowledge consists of knowledge of how to operate GSP. Content knowledge includes all of the mathematics learned prior to participating in the research activities. Lastly, problem solving knowledge equates to an understanding of heuristic actions (mathematical and technological), their trajectories, and their effectiveness and associated errors as revealed through student work in the completion of the research activities (Özgün-Koca, Meagher, & Edwards, 2010; Schmidt, Baran, & Thompson, 2009).

Problem Solving and Dynamic Mathematical Environments

Through significant development in the 1980's and 1990's (seminal literature from the era cited here), the literature reveals that numerous researchers treat and investigate problem solving as an independent content area to be studied and applied (e.g., Cobb, Wood, & Yackel, 1991; Evan & Lappin, 1994; Lester, Masingila, Mau, Lambdin, dos Santon, & Raymond, 1994; Olkin & Schoenfeld, 1994; Schifter & Fosnot, 1993; Schoenfeld, 1994; Stanic & Kilpatrick, 1989; Van Zoest, Jones, & Thornton, 1994). As a field of study, problem solving continues to be investigated in respect to: students/students and teacher/students interactions; mathematical dialogue and socially mediated understanding; students employing multiple heuristics; teacher questioning techniques; and encouraging students to make generalizations.

From the perspective of learning, Schoenfeld (1994) recognizes problem solving as a sense-making tool leading to generalization and Lester et al (1994) posit that teaching and learning through problem solving helps students develop conceptual understanding through mathematical engagement involving creating, conjecturing, exploring, testing, and verifying. Cobb, Wood and Yackel (1991) suggest that, through problem solving, students cognitively reorganize concepts involved in the activity. Summarily, problem solving is recognized as a skill or experience that improves learning (NCTM, 2000).

From the psychological, social, and affective domains, problem solving develops students' mathematical self-confidence (Schifter & Fosnot, 1993) and acts as a conduit through which students communicate, construct, evaluate, and refine both individuated theories and those of others (NCTM, 2000). Stanic and Kilpatrick (1989) consider how problem solving can motivate student learning and Olkin and Schoenfeld (1994) consider the affective nature of problem solving leading to engagement, time on task, and overall enjoyment.

From the perspective of problem solving as a process, students are to investigate their individual problem solving heuristics as well as those of others in order to better understand and learn about problem solving (Carpenter, 1989; Stacey & Groves, 1985; Thompson, 1985). Thus, problem solving has evolved into an independent domain of knowledge to be either studied or experienced.

There are a variety of tools and accessories students have available to them to aid in mathematical problem solving. Lately, the growth of technology has provided a major supplement for students solving mathematical problems (Lee & Hollebrands, 2006). In order for technology to have an effective role in student problem solving, students should be cognizant of these available technological resources and the problem solving processes they employ when using them.

Dynamic geometry environments (DGEs) (e.g., The Geometer's Sketchpad (GSP), Cabri, GeoGebra, and Cinderella) allow concretizations of mathematical concepts through dynamically interconnected representations and afford users the ability to transform and explore mathematical relations and properties and develop geometric conjectures (Hoyles & Noss, 2003). Common features associated with all DGEs include: a set of primitive Euclidean elements; the ability to use these elements to construct other geometric objects; the ability to use transforming, measurement, and calculation tools on constructed objects; and the ability to drag, animate, and hide/show objects and measurements to explore relations among constructed objects (Hollebrands & Lee, 2012).

Through user action, DGEs can articulate and reinforce mathematical relationships through instantaneous feedback and characteristics can be dynamically explored (Arzarello, Micheletti, Olivero, Robutti, Paola, & Gallino, 1998; Goldenberg & Cuoco, 1998). Beyond simply improving on pre-technical geometric manipulations, DGEs allow users to construct, interact with, and investigate mathematical relations and ideas in novel forms (Pea, 1985; Ruthven, Hennessy, & Deaney, 2008).

Extending from DGEs to tools that are developed to consider all mathematical fields in a more balanced fashion (not overly focused on geometry), dynamic mathematical environments (DMEs) are technological environments (hardware and software) with a mathematics embedded infrastructure such that users may construct or interact with dynamic mathematical representations (Dick & Hollebrands 2011; Lopez-Real & Leung, 2006). (Notably, depending on how it is used, most DGEs are also DMEs.) Students use DMEs as a technological mathematical manipulative in order to gain deeper and more interconnected mathematical understanding and to students conceptually navigate through a representation system (Duval, 2006). DMEs allow students to dynamically visualize and portray mathematical concepts and relations and to manipulate concepts through dynamically linked, multiple representations. Using a DME, students can explore behavior of particular geometric figures, investigate relational patterns, or create animations to demonstrate the mathematical concepts (Finzer & Bennett, 1995).

DMEs, such as The Geometer's Sketchpad, have become a key technological tool students can use for mathematical problem solving. They provide students the opportunity to create and manipulate constructions and investigate multiple dynamic representations simultaneously (Garofalo et al, 2000). Students can integrate geometry, algebra, and other mathematical fields and investigate interconnections among these. Using a DME allows students to more readily view and interact with different representations of the same mathematical idea (Lee, Harper, Driskell, Kersaint, & Leatham, 2012). Lee and Hollebrands (2006) give an example that students may understand certain mathematical ideas using a DME because they can manipulate objects or constructions directly through dragging as compared to merely input commands. GSP, and DMEs more generally, allow researchers to approach investigations of student understanding of translations by analyzing the ways in which students perform translations in dynamic environments (DePeau & Kalder, 2010).

A primary difference between DMEs and static multiple mathematical representations is that DMEs allow users to dynamically interact with (through dragging and other techniques) any particular representation, thus affecting student awareness of, and learning through, multiple interconnected representations (Arzarello, et al, 1998; Arzarello, Olivero, Paola, & Robutti, 2002; Baccaglini-Frank, & Mariotti, 2010). Static representations, typically found in textbooks, are incapable of being dynamically altered by the user and the connections between multiple representations are often unseen. This dimension may affect how students learn; they may learn differently when working in DMEs versus more traditional static instructional materials. Paralleling processes involved in translating between static representations, the downfall for any DME is that syntactically interacting with the technology can lead students to memorize keystrokes and action rather than fully understand what the actions mean and produce. Students who memorize DME commands can produce correct results without being able to explain why and how the commands work. Later discussions will investigate if the errors performed by students when translating between static representations are similar to, or different from, errors produced when students work in DMEs.

Mathematical Errors

Students make a variety of errors when performing any mathematical task. Errors include, but are not limited to, misinterpreting expressions, misapplying mathematical properties (e.g., field axioms or order of operations) (Finzer & Bennett, 1995). In respect to translating among mathematical representations, two distinct types of errors have been recognized in the literature: manipulation errors include incorrect arithmetic or algebraic calculations or variable misinterpretation and conceptual errors include the introduction of an incorrect constraint (errors of commission) or the overlooking of critical constraints (errors of omission) (Bell, Brekke, & Swan, 1987; Kerslake, 1981; Preece, 1993).

As previously mentioned, syntactic elaboration occurs when a student manipulates a problem incorrectly but understands overall the mathematical concepts, and semantic elaboration occurs when the student is able to work all the steps of a problem individually but hasn't fully grasped the main concept (Kaput, 1987a, 1987b). Strong parallels can be seen between syntactic thinking and manipulation errors and semantic thinking and conceptual errors. In this paper we codify these distinctions by denoting syntactic errors and semantic errors.

Since all translation processes can be captured through the activities of interpretation, action, and assessment, it can be argued that all errors must be definable by these three activities (Adu-Gyamfi, & Bossé, 2014; Adu-Gyamfi et al., 2012, 2015). While a great deal of previous literature exists in the realm of mathematical errors, most of this literature simply investigates whether or not a student successfully performs a task; very little literature specifically defines or classifies types of errors (Adu-Gyamfi et al., 2012, 2015). Student errors can take various forms. An error can be as minute as overlooking a plus or a minus sign when solving an algebraic equation or misusing the order of operations (a syntactic error). In contrast, mathematical errors can be as complex as failing to recognize the interconnection of polynomials in symbolic and graphical forms (a semantic error).

Altogether, defining error types is essential to understanding types and frequencies of student errors. This is all the more so necessary when investigating student interaction in novel DMEs. This study seeks to capture and define error types observed through student work as they solve problems in a DME and extend the literature in respect to both error types and student activity in DMEs.

Methodology

Participants

Participants in this study were undergraduate students with intended majors of mathematics, computer science, statistics, physics, elementary education and sustainable development at a university in the southeastern United States. Participants were recruited based on their major, anticipating that students in these majors would possess the mathematical background commensurate with the activities presented to them. The requisite mathematics associated with the research activities did not surpass what would traditionally be encountered in high school mathematics – particularly for students who took a precalculus and a discrete mathematics course. Mathematical topics in this study included elementary metric geometry and Boolean operations.

Three of the students had previously had a university geometry course in which GSP was extensively used. The remainder of the students had no previous involvement with GSP. In order to learn the basics of Boolean operations within the GSP environment, all study participants were provided two days of researcher-directed, in-class training activities supplemented with online video tutorials, instruction, and activities that were provided for the duration before the performing of the research activities.

Research Activity

One week after the two days of training, participants returned to attempt the research task. While there was no attempt to measure whether the training was effective, nor was there any pre-assessment given to assess students' knowledge of GSP, how to use Boolean operators in GSP, or of Boolean operators or geometry in general, these dimensions were addressed in significant detail in the training sessions. The research activity was defined as: In Geometer's Sketchpad, construct an independent and movable triangle and circle; as you move the figures around the screen, use Boolean operations to dynamically report whether the triangle is inside the circle, the circle is inside the triangle, the figures overlap, or the figures are disjoint. The proposed results of the research activity are provided in Figure 2. In a computer lab, participants worked independently and were allotted three hours to complete the activity.

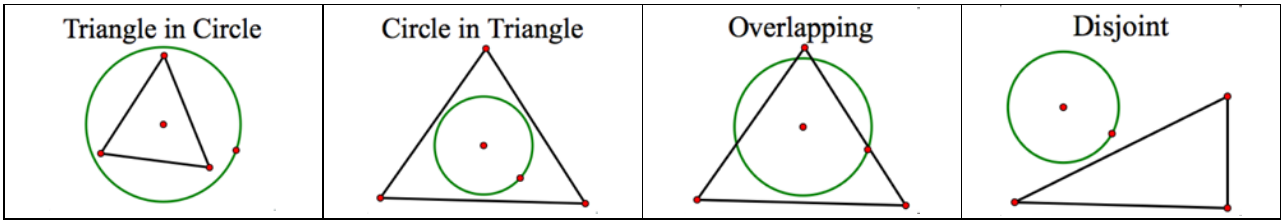

Figure 2. Proposed results of the research activity.

In order to accomplish this task, participants needed to construct and measure geometric figures and employ Boolean operators and comparisons such as: AND, OR, NOT, greater than, less than or equal to, equalities, and inequalities regarding various conditions and measurements in the sketch. With these inherently necessary components, five or six conventional combinations of these components could be anticipated as possibly leading to successful heuristics. However, while a far greater number of unanticipated heuristics could also prove successful, these, too, would necessitate some combination of these components. Thus, these tasks could be characterized as open-ended with somewhat predictable successful heuristics.

Data and Analysis

As participants worked on the research task, Camtasia Studio was used to record every aspect of their work on the computer, all their verbal conversation between themselves and researchers, and their facial expressions synchronized to the video of their work as they participated in a task-based interview (Goldin, 2000). Additional data was obtained through student written work, if they used scratch paper.

As will be later seen in Figure 3, the complexity of interpreting final student work in this project necessitated analyzing student actions through step-by-step recordings as they completed the task. Through these recordings, particular dimensions could be observed and analyzed including: each step taken in their work; the reasoning behind their actions; actions performed and altered or erased; and the iterative or repetitious nature of processes. Many of these dimensions would have been lost by only considering final products.

The data collection, observation, analysis, and synthesis used in this study followed well-established techniques of qualitative research, albeit with a twist that is discussed below. The established practices of data analysis proceeded through sequential phases. First, data was transcribed. Second, Camtasia data and transcripts were coordinated and reviewed to identify themes and categories (Bogden & Biklen, 2003; Creswell, 2003) and data was sorted into those categories. Questions about the data were documented, possible explanations for these questions were hypothesized, and confirming or disconfirming evidence was systematically investigated (Strauss & Corbin, 1994). Third, data was coordinated across data involving all participants. The data was then re-examined and the codes were assigned to small units of data. Common themes within, and across, the differing participants were then identified. These coding structures were compared and differences reconciled; these processes resulted in the refinement of initial codes. Researchers were able to employ the process of check-coding (Miles & Huberman, 1994) and thereby reach consensus on the analysis of all transcripts. The reconciliation enabled the researchers to clarify their thinking and to sharpen each code's definition.

The atypical nature of this research methodology resides in that the preceding analytical sequence was performed iteratively comprising a total of four rounds of analysis and synthesis. While each round produced additional findings, they also provided opportunity to repeatedly verify previous findings of themes and employment of codes. To initially frame this study, the interconnections of three domains of knowledge (mathematics, technology, and problem solving) were considered. The second phase of data analysis considered processes involved in problem solving (interpretation, activity, evaluation). The third phase of analysis considered the dimension of student interaction with the domain of knowledge (syntactic versus sematic). The codes employed during these analyses are provided in the discussion section of this paper.

While data was analyzed for all ten participants involved in this study, this paper focuses on data from only two participants. Data from the other participants revealed little more regarding the central investigation associated with this paper: defining student error types in respect to problem solving in DMEs. It is hoped that additional findings regarding error sequencing will be determined through future studies.

Notably, the Problem Solving Error Identification Framework (PAIF), defined later in this paper, categorizes all activities involved in the problem solving process and simultaneously captures those that are errors. This framework resulted from the iterative analyses above. After the analyses led to findings and the development of the PAIF, the PAIF was then employed again as a tool to further analyze and synthesize findings.

Initial findings from these analyses are provided in the following section with more detailed results in the discussion section.

Findings

The findings from student work are multidimensional, and some were unanticipated. It was anticipated that student work could be recognized to contain errors that could be associated with incorrect understanding or applications of mathematics or technology. However, a number of additional dimensions were recognized to characterize error types. These characteristics are discussed below in the order in which they were discovered (i.e., domain errors, process errors, and interaction errors) through numerous iterations of analyzing student work on the activity.

Since a number of different dimensions of errors are detailed below, it is valuable to preface the following subsections with a descriptive nomenclature. These errors will be again defined in closer proximity to providing examples from student work of each error type.

Students make *Domain Errors* when they misunderstand, misapply, or incorrectly perform activities associated with one of the three domains of mathematics, technology, or problem solving. For example, there may be a mathematical miscalculation, a misunderstanding in how to make the technology perform a particular action, or a cyclical use, rejection, and reuse of a problem solving heuristic that has previously proven unsuccessful. Domain errors can simultaneously be one or both of process errors or interaction errors.

As students address any domain, they are either interpreting the domain, performing some activity on the domain, or evaluating notions associated with the domain. *Process Errors* denote student errors in any one of these three processes. For instance, a student may misinterpret some mathematical notion, incorrectly perform some technological task, or incorrectly evaluate his results in respect to an anticipated outcome.

Students interact with a domain in two ways: syntactically (locally) or semantically (globally). *Interaction Errors* occur when students either: work with precise content attributes and characteristics, but misinterpret global concepts (syntactic error) or understand global content concepts, but misinterpret precise attributes and characteristics (sematic error). For instance, if a student syntactically (locally) understands that, for $f(x) = ax^2+bx+c$, a, b, and c are real-valued coefficients, that x is an independent variable, and that he can solve this equation for x, but he does not understand that this is a quadratic function with determinable roots and understood graphical behavior, he may perform a syntactic error. Conversely, if a student semantically (globally) understands that $f(x) = ax^2+bx+c$, represents a quadratic function with determinable roots and understood graphical behavior, but cannot correctly substitute values for a, b, and c, or solve for x using the quadratic formula, he may perform a sematic error.

The following demonstrate simple examples of these error dimensions. These examples are snippets taken from larger transcriptions of student work. The fuller transcriptions are provided in later discussions to provide more context for findings and discussion. Notably, researcher notes are included in the transcripts provided in this document. These notes are interpretations of student behaviors and thinking and comments regarding the potential of the selected heuristic and denoted as italicized text within brackets.

To assist the reader in understanding student work, simplified figures representing actual student work accompany the transcripts. Actual student work (before it is cleaned up by hiding all but the necessary final elements) is busy and cluttered on the screen (including geometric figures and constructions, measurements, Boolean expressions and comparisons, calculations, and text, and much of this is overlapping), and would only be interpretable by readers with significant experience in GSP and Boolean operations. An example of typical student work is seen in Figure 3.

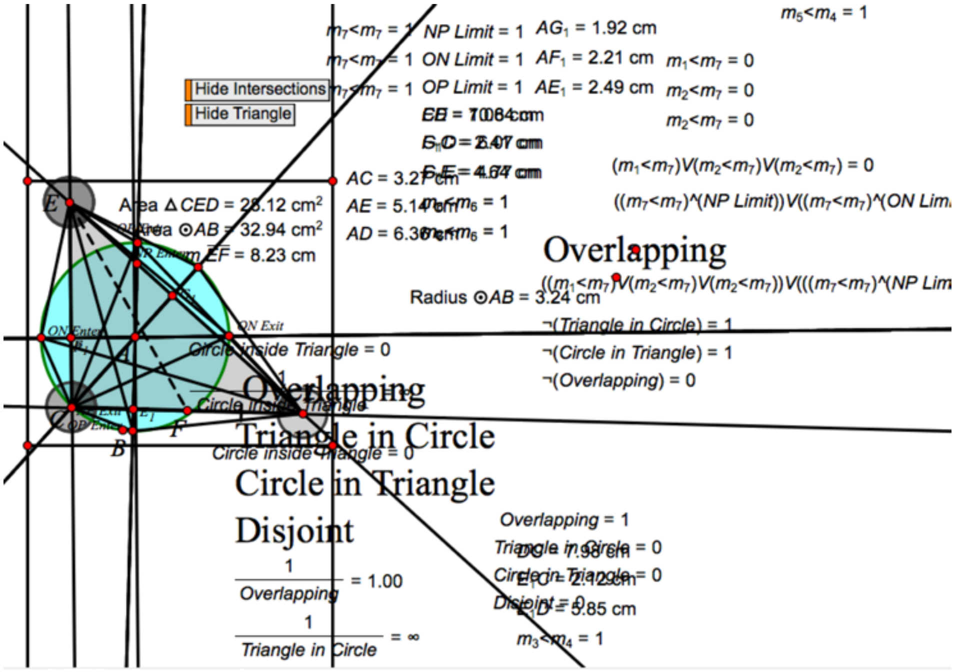

Figure 3. Example of student work before cleanup

Domain Errors

As the domain of mathematics has defined processes to be understood and followed for its proper use, so too does the domain of any technology and software platform. Misunderstandings of the technological environment or misapplications of the technology tools in the environment constitute an error in the domain of technology. Student work was found replete with errors that could be clearly associated with misunderstandings or misapplications of either mathematics or the available technology. However, students also made errors that could be associated to the realm of problem solving – neither centered in mathematics nor in technology. This led to the recognition of problem solving as a distinct domain of knowledge. (This is addressed more thoroughly throughout.) Below are examples of domain errors.

The error in the work of Student 1, below, albeit minor and readily rectified, is technological in nature. He is unsure of how to make Sketchpad perform the arc construction he wanted.

> When constructing the arc on the circle to then make a circle segment, Student 1 repeatedly [*and erroneously*] constructs major arcs rather than minor arcs.

The following referenced error in the work of Student 2 is an error associated with insufficient understanding and application of appropriate mathematics and constitutes a mathematical domain error.

> Student 2 [*incorrectly*] attempts to solve this problem using the triangle medians. [*His technique will not be helpful.*]

Since the research activity involved work in a DME, mathematics and technology were thoroughly and necessarily intermingled in the activity. While some errors in student work were occasionally difficult to discern regarding being born from misunderstandings of mathematical content versus technology use, most were readily discernable.

Notably, errors were not categorized as problem solving domain errors simply because they did not fit well into the camps of mathematical or technological domain errors. Rather, they were deemed problem solving errors when: the problem seemed insufficiently understood; mathematical, technological, or problem solving steps were taken which the researcher knew would prove unfruitful in solving the problem; students left potentially fruitful heuristic paths for less hopeful ones; or students iteratively returned to heuristics that had earlier proven unsuccessful. As such, the following work from Student 1 demonstrates problem solving domain errors.

> Student 1 [*seemingly absentmindedly*] repeats many of his previous mistakes, redoing his previous processes. Upon realizing that he was again proceeding down the same path, he abandons this and starts constructing segments from the center of the circle to the vertices of the triangle. [*A technique with potential.*] After a while, he deletes these segments but still retains the intersections with the triangle and the circle and attempts the same approach above for the third time.

Process Errors

After a recognition that errors are made in each of the domains of knowledge, it was found that some errors could be situated in distinct problem solving processes. As previously stated, these activities include: interpretation, activity, and evaluation. Examples of these dimensions follow.

Interpretation errors. Interpretation errors occur as students incompletely or incorrectly interpret any of three domains: the problem, the mathematics, or the available technology tools. Below, Student 1 is having difficulty mathematically interpreting what he is observing. His subsequent mathematics errors (reported later) evolve from his inability to mathematically interpret what he is observing.

> Having successfully completed the Boolean calculations to determine if either of two sides of the triangle intersects the circle and realizing that he is having difficulty to determine this for the third side, Student 1 repeatedly manipulates the circle and triangle to investigate the behavior in respect to the third side. When asked what he was doing, he responds, *I'm not sure what is happening here. I am moving the circle and triangle to see what is happening on this side. I'm not sure why it seems that this third side is different from the other two.*

Below, the work of Student 1 demonstrates his misinterpretation of the problem activity. He correctly performs what he wishes to accomplish, but realizes that his understanding of the problem is insufficient. This misinterpretation of the problem leads to numerous errors.

> [*When the work of Student 1 transitions from geometric constructions to Boolean comparisons, he begins to struggle his global understanding of the activity seems incomplete.*] He measures [*unnecessary*] distances and creates Boolean comparisons to determine whether these measures are equal to a randomly placed radius. Realizing

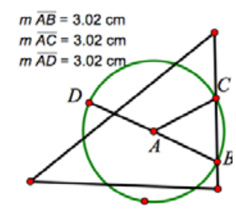

Figure 4

> that this technique could not produce the results he hoped for, he deletes the Boolean comparisons but keeps the measurements of the distance from the center of the circle to the intersections of the triangle segments and the circle [*although the measurements had no potential to lead to a solution, because each measurement was simply another radius*]. (See Figure 4.)

Activity errors. In the process of translating between static mathematical representations, translation errors occur after the source representation is properly interpreted and during the actual acts of producing the target representation. In problem solving activities employing DMEs, activity errors occur as students attempt to perform the operations or construction necessary to carry out the task. Below, Student 2 has correctly interpreted the problem both mathematically and technologically. However, his current errors are in his application of technology tools to do as he wishes. His activity error is in respect to performing a technological task.

> Student 2 repeatedly attempts to create a necessary Boolean operation, but keeps forgetting to select the correct elements for it to work properly. Minutes later, he [*accidentally*] selects TRACE POINT [a *technological error*] without intentionality. He tries to merge text to a point but doesn't realize he has two points highlighted and it does not work as he hopes [*a technological error*].

In the following transcript, while the use of a coordinate grid is a vaguely possible heuristic, the use of lattice points has no value whatsoever. Nevertheless, although Student 2 inserts a coordinate grid into his work, he never makes use of any aspect of it. Thus, he has performed an act that has no value in the problem solving process and one that may indeed hinder his progress. This is a problem solving process activity error.

> Student 2 begins attempting to make use of the coordinate grid and tries to make the points fall on exact lattice points to see how the triangle and circle interact. [*The background coordinate grid is seemingly hindering his problem solving. When elements are dragged around the screen, users are not guaranteed that these shapes will stay precisely on lattice points. (There is a SNAP POINTS feature available to accomplish this, although, in this scenario, this would also have been of little value.)*]

Evaluation errors. Evaluation errors take one of two forms: either students do not attempt to assess that their work performs appropriately as tasked or the assessment is incomplete. Either potentially leads to task completion errors. While both students in the following transcripts fail to adequately assess that their work is performing correctly, the forms of their errors differ somewhat. Student 1 fails to take advantage of a known option in Sketchpad that would have allowed him to investigate what was going wrong and give him insight into how to correct it. More precisely, the work from Student 1 demonstrates evaluation errors in respect to both mathematics (he only drags elements about the screen rather than checking results mathematically) and problem solving (his dragging of elements seems to indicate his lack of certainty regarding the nature of the problem and little effort is made to verify that he understands the task).

> Eventually, Student 1 [*again*] constructs segments between the center of the circle and the vertices of the triangle. [*Although he could have SHOWN ALL HIDDEN work to be able to investigate the error, he seems reluctant and rather prefers to continue manipulating and dragging the triangle and circle about the screen.*]

Below, Student 2, rather than assessing his work and determining what is incorrect, allows his confusion to send him toward other heuristics – never fully knowing if earlier paths would have proven successful.

> Student 2 beings to compare the area of the circle with the radius and one midsegment of the triangle. During this approach, he seems confused regarding what elements he should be comparing. He constructs all the coordinate points of the triangle and continues to construct the area of the triangle, [*but hesitates in doing something meaningful and evaluative with these calculations*]. (See Figure 5.)

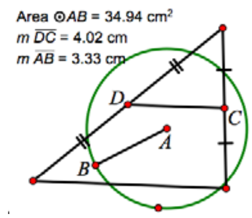

Figure 5

Interaction Errors

With the recognition of domain and process errors in place, another iteration of analysis led to the observation of another dimension of error types. Students interact differently with the problem and domain. While all students interact with the domain of knowledge, some do so through a full understanding of their global natures and others do not understand the global nature of the domain. Thus, interaction errors take one of two forms: syntactic and semantic errors. Syntactic errors occur when students perform problem solving operations and actions but make missteps due to not fully understanding the global picture of the domain of knowledge. Semantic errors occur when students globally understand the content areas but make minor and intricate errors regarding the domain. Examples of syntactic and semantic errors follow.

> [*Student 1 seems to well understand the activity from a geometric standpoint, the technology, and the problem.*] He creates a triangle and a circle and constructs all six intersection points for when the figures overlap. [*This technique will eventually fail, as the elements will not exist under other conditions and will become unusable in constructions or calculations*]. He determines the area of the circle segment created with respect to one side of the triangle and observes how this changes as the circle moves positions. (See Figure 6.) He extends this technique to consider the three sides of the triangle.

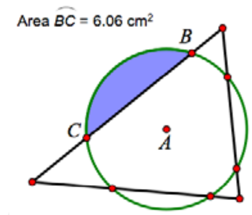

Figure 6

Above, Student 1 appears to globally understand the problem; however, he struggles with the particulars of mathematical and technological details that will produce the desired results. This represents a semantic interaction error. Below, Student 2 is able to use mathematics and the

Boolean operators to get some results. However, his work demonstrates a lack of global understanding regarding how the mathematics and technology interact to produce the Boolean operations.

> Student 2 [*believes he*] discovers conditions indicating when the circle is completely inside the triangle. He uses Boolean operators to report that, if the distance from the centroid to the center of the circle is less than the radius of the circle, then the Boolean operator should show “1”. (See Figure 7.) However, the calculation shows =1 under the other [*unwanted*] conditions. [*He does not seem to have a global understanding of the technology.*]

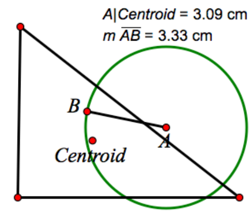

Figure 7

Error Type Organization

Although each error type was previously defined, the observed interactions of these error types warrant discussion. As was recognized through the research methodology involving iterative processes of analysis and synthesis, some errors could be encoded by multiple error types. While no error was simultaneously in two different domains, processes, or interactions, it was noted that interaction errors and process errors could occur in any domain of knowledge. Indeed each error could be situated in one each of the dimensions of domain, process, and interaction. Thus, for instance a student could make an error that could be denoted as a syntactic interpretation error in mathematics or a semantic evaluation error in problem solving or another error denoted as a semantic activity error in mathematics.

The interaction among, and overlapping nature of, error types is demonstrated in greater detail in the following discussions. However, significant in these immediate findings is that three types of errors were discovered in student work regarding problem solving in a DME: domain (mathematics, technology, and problem solving), process (interpretation, activity, and evaluation), and interaction (syntactic and semantic). Additionally, it is noted that interaction, process, and domain errors can all work together. We address these findings in the following discussions.

Discussion

The preceding findings define error types discovered through observations of student work and communication in respect to the research task. These findings lead to additional dimensions that warrant attention. These follow below in respect to the Problem Solving Error Identification Framework (PAIF) and additional findings through analyses of fuller transcripts.

Problem Solving Action Identification Framework

The preceding findings define a number of error categories in student work in DMEs. These include: domain, process, and interaction errors. Table 1 synthesizes all of these error categories into a framework that can be used to investigate student errors when problem solving in DMEs.

While the Problem Solving Action Identification Framework (PAIF) synthesizes the findings of error types and characteristics mentioned previously, additional dimensions in the framework warrant attention. These are discussed briefly.

First, the PAIF can be seen to encapsulate all actions associated with problem solving. Depending on how the action is performed, some of these actions can be deemed as errors.

Thus, for future studies, the PAIF can investigate all problem solving actions, whether correct or erroneous.

Second, the PAIF recognizes problem solving as a domain of knowledge equivalent to mathematics and technology. As with other content areas, problem solving can be performed or investigated as a field of study. Thus, errors can be made in the domain of problem solving as well as in other domains. Problem solving errors can be in conjunction with, or independent of, other domains within the problem solving task.

The PAIF also articulates that problems must be interpreted in order to be solved. Altogether, each domain of knowledge must be interpreted in order to suggest possible heuristics. Indeed, interpreting a problem must be the first step to solving it. As seen in some student work, students often flail away at solution strategies without adequately understanding the problem or in an attempt to interpret it more thoroughly. (A more complete discourse on problem solving is articulated in later discussions.)

Table 1. The Problem Solving Action Identification Framework

Problem Solving Action Identification Framework						
Process	Domain of Knowledge					
	Mathematics		Problem Solving		Technology	
	Syntactic	Semantic	Syntactic	Semantic	Syntactic	Semantic
Interpretation	Working with local mathematical attributes and characteristics, but misinterpreting global mathematical concepts.	Understanding global mathematical concepts, but misinterpreting local attributes and characteristics.	Working through local problem solving strategies, but misinterpreting the larger problem at hand.	Understanding the global problem at hand, but misinterpreting local problem solving components.	Working with local technological attributes and characteristics, but misinterpreting larger technological aspects of the environment.	Globally understanding the technology, but misinterpreting local aspects of the technology.
Activity	Working with local mathematical attributes and characteristics, but incorrectly applying global mathematical concepts.	Understanding global mathematical concepts, but misapplying local attributes and characteristics.	Working through local problem solving strategies, but misapplying global problem solving concepts.	Understanding the global problem at hand, but misapplying local problem solving components.	Working with local technological attributes and characteristics, but misapplying global technological aspects of the environment.	Globally understanding the technology, but misapplying local aspects of the technology.
Evaluation	Considering local mathematical attributes and characteristics, but incorrectly assessing global mathematical concepts.	Understanding global mathematical concepts, but incorrectly assessing local attributes and characteristics.	Considering local problem solving strategies, but incorrectly assessing global problem solving concepts.	Understanding the global problem at hand, but incorrectly assessing local problem solving components.	Considering local technological attributes and characteristics, but incorrectly assessing global technological aspects of the environment.	Globally understanding the technology, but incorrectly assessing local aspects of the technology.

Since the totality of the domains of knowledge involved in these tasks include mathematics, technology, and problem solving, the PAIF must, by definition, contain all possible domain errors. This will greatly assist future studies employing this framework in the investigation of student work in respect to problem solving and DMEs.

Third, previous research has demonstrated that all processes associated with translations between static representations can be encapsulated in three activities (i.e., interpretation, translation, and evaluation) and that all possible errors could be ascribed to one of these three activities (Adu-Gyamfi, & Bossé, 2014; Adu-Gyamfi et al., 2012, 2015). Similarly, this study claims that all problem solving activities and subsequent errors can be recognized in one of the three activities: interpretation, activity, and evaluation. Since this dimension captures all possible activity errors, the PAIF will be a valuable tool in future investigations.

Fourth, reasons can be ascribed to why students make particular errors. Some students may possess a global understanding of the task through one or more of the domains of knowledge, but make errors in respect to the domain(s) (semantic errors). Thus, a global awareness of the problem task is not sufficient to avoid committing errors. Other students may not possess a global understanding of the task or of one or more domains associated with the task, but may perform domain tasks correctly (even though these domain tasks do not necessarily help to solve the problem). Errors are then born from students correctly performing domain tasks that either lead to no positive results or are disjointed and disconnected from potentially successful heuristics (syntactic errors). Thus, performing tasks correctly in the associated domains during problem solving is not sufficient to avoid the commission of errors.

This study does not go so far as to imply that semantic and syntactic interactions within problem solving are either complementary or all inclusive. However, with this advancement in the notion of student interaction in problem solving, it is hoped that future research will provide either verification of these interactions or catalog additional ones.

Fuller Transcripts and Additional Findings

After synthesizing the findings into the formation of the PAIF, this framework was employed to again analyze student work. In this process, an additional layer of application was discovered. Although the PAIF was designed to capture and denote errors and error types, it seems to also capture all actions involved in problem solving – some of which are errors. This parallels the fact that all processes and errors associated with translations between static representations are encapsulated in three activities. The transcripts below demonstrate the coding structure infused in transcripts of student work. The codes denote all the actions involved in the problem solving process. Although this is not the primary purpose for this research, these action codes (whether or not the actions constitute errors) provide more insight into student work and thinking. Employing the PAIF to investigate all actions associated with problem solving processes – beyond principally investigating errors – may be a valuable tool in future investigations.

Processes	Domain of Knowledge					
	Mathematics		Problem Solving		Technology	
	Syntactic	Semantic	Syntactic	Semantic	Syntactic	Semantic
Interpretation	IMY	IME	IPY	IPE	ITY	ITE
Activity	AMY	AME	APY	APE	ATY	ATE
Evaluation	EMY	EME	EPY	EPE	ETY	ETE

Table 2. Discrete and Overlapping Error Types

Below, fuller transcripts are provided with embedded research commentary and PAIF coding in order to provide fuller context of previous findings and develop further ideas associated with PAIF. The PAIF coding (Table 2) works as follows: codes appear as three alphabetical digits. The first digit represents the first letter from the problem solving process from which the error originated (i.e., I=interpretation; A=activity; and E=evaluation). The second letter represents the first letter of the domain of knowledge from which the error originated (i.e., M=mathematics; P=problem solving; and T=technology). The last letter indicates the second letter of the interaction from which the error originated (i.e., E=semantic and Y=syntactic). Thus, the code AMY would represent a syntactic error occurring during the activity process, in the mathematics domain.

Additional sub-codes aid in the explanation of student activity. The symbols "+" and "~" are added to some codes. The use of "+" indicates that a positive action or change occurred. The symbol "~" indicates an omission of a process that could have proven helpful. While all codes indicate problem solving actions (working with domains, problem solving processes, or interactions), processes in the codes that are in a bold font are actions that are errors of either commission or omission.

Student 1.

Student 1 immediately attempts to apply Boolean operations [*but does not realize that he first needs to measure a length*]. [~ITE] He selects line segments [*rather than their measurements*] and tries [*unsuccessfully*] to use the Boolean commands. [~ETY]

[*He seems to well understand the activity from a geometric standpoint, the technology, and the problem.*] He creates a triangle and a circle and constructs all six intersection points for when the figures overlap. [*This technique will eventually fail, as the elements will not exist under other conditions and will become unusable in constructions or calculations*]. He determines the area of the circle segment created with respect to one side of the triangle and observes how this changes as the circle moves positions. (See previous Figure 6.) [AME] [*Since this area of the sector is dependent upon the intersection points existing, this technique will fail when the intersection points do not exist.*] He extends this technique to consider the three sides of the triangle.

He does not take into account cases in which the circle and triangle overlap on only one or two sides of the triangle. [~AMY, ~APY, ~EMY] To construct a circle segment, he repeatedly [*and erroneously*] constructs major arcs rather than minor arcs. [ITE, ATE, EMY]

He then attempts a new heuristic on a new sketch page – reserving his last page in Sketchpad in case he needed to refer back to it. [*Abandoning the consideration of areas of circle segments that may in some cases not exist*], he begins to investigate the distance from center of the circle to intersections of triangle with the circle. (See previous Figure 4.) [*While investigating the distance from the center of the circle to particular points is potentially fruitful, each of these intersection points is on the circle and all the distances are equal to the radius of the circle. A more correct method would be to determine the distance from the center of the circle to the vertices of the triangle.*]

His work transitions from geometric constructions to Boolean comparisons [*a necessary aspect of this activity*] and [*his understanding of the big picture of the activity seems to be less complete.*] He creates Boolean comparisons to determine whether these measures from the center of the circle to the intersections of the circle and the triangle sides are equal to a randomly placed radius. [*They are.*] He [*unsuccessfully – as this can never be the case*] attempts to discover cases where these measurements would not be equal to the radius. [AMY] [*He was seemingly misinterpreting "exist and not exist" with "true and not true."*] [IMY, ITY, ~EMY, ~ETY] Realizing that this technique could not produce the results he hoped for, he deletes the Boolean comparisons but keeps the measurements of the distance from the center of the circle to the intersections of the triangle segments and the circle [*although the measurements had no value, because each measurement was simply another radius*]. [~EMY] [*Working too rapidly*] [APY, ~EPY], he frequently selects incorrect elements and needs to correct his actions. [ATY, ~ETY, ~EPY]

Continuing his work, he [*absentmindedly*] repeats many of his previous mistakes, redoing his previous processes [APY, ~IPE, ~EPE]. Upon realizing that he was again proceeding down the same path, he abandons this and starts constructing segments from the center of the circle to the vertices of the triangle. (Figure 8.) [*This could have been a successful technique to determine if the triangle is completely inside the circle.*] After a while, he deletes these segments [~EMY] but still retains the intersections with the triangle and the circle and attempts the previous approach for the third time. (See previous Figure 4.) [IPY, ~EPY]

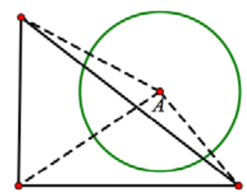

Figure 8

Incorporating numerous Boolean calculations, he attempts to make the word "overlap" appear when appropriate. He successfully makes this work for two sides of the triangle; in respect to the third side, he seems to get confused with the number of Boolean calculations and misapplies some. [AMY, ATY] He seems to understand the problem – being aware of the error in having the overlap case report correctly – but is having difficulty locating the error in the Boolean commands for the third side. Attempting to locate the error, he repeatedly manipulates the circle and triangle to investigate the behavior in respect to the third side.

Eventually, he [*again*] constructs segments between the center of the circle and the vertices of the triangle. (See previous Figure 8.) [~EPY] [*Although he could have SHOWN ALL HIDDEN work to be able to investigate the error, he seems reluctant and rather prefers to continue manipulating and dragging the triangle and circle about the screen.*] [~EMY, ~EPY]

[*While he hasn't fully abandoned the radius approach*] [~EPY], he moves forward and measures the triangle medians. (Figure 9.) [AMY, ~IMY, ~EMY] After dragging triangle vertices around the screen for a few minutes [*he seems to be contemplating his next moves*], he creates a series of Boolean operators that correctly depict the overlap feature. He moves on to

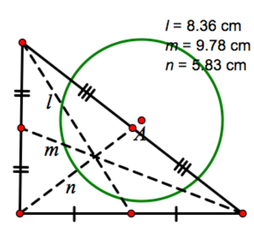

Figure 9

consider the case such that the triangle is inside the circle. He [*correctly*] measures the distances from the center of the circle to the vertices of the triangle and [*correctly*] creates Boolean operations to compare the radius length with the distance from the center of the circle to each triangle vertex. (See previous Figure 8.) Moving to the next case, he tries to simultaneously examine the cases where the triangle and circle are disjoint and the where the triangle is inside the circle. His commands do not work together the way he anticipates. [ATY] [*While this technique demonstrates that he recognizes the differences in the two cases, this technique will not prove to be valuable. It is unclear whether his technique is born from a lack of understanding of the full activity or the precise interconnection of mathematical concepts.*]

[*Noticeably rushing, due to running out of time*], he constructs the perpendiculars from the center of the circle to each side of the triangle. (Figure 10.) [~IMY, ~EMY] He measures these lengths and observes how these lengths change as the circle is moved from inside the triangle to being disjoint with the triangle. [*Time runs out and he stops the activity at this point.*]

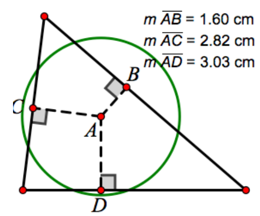

Figure 10

Student 2.

Student 2 repeatedly attempts to apply a Boolean operation, but keeps forgetting to select the correct elements to do so. [ATY, ~ETY] Minutes later, he [*accidentally*] selects TRACE POINT. [ATY] He attempts to merge text to a point but doesn't realize he has two points highlighted. [ATY] For some [*indiscernible*] reason, he adds a coordinate grid to the background and uses the grid in his following actions. [~IPE, ~EPE]

He begins to compare the area of the circle with the radius and one midsegment of the triangle. (See previous Figure 5.) [~IPE, ~EPE] During this approach, he seems confused regarding what elements he should be comparing. He determines the coordinates of the triangle and constructs the area of the triangle, but hesitates in doing something [*meaningful*] with these calculations. [~IPY, ~IMY, ~EMY] [*He seemingly understands the problem in the global sense, but he incorrectly assesses precise problem solving components such as squaring the radius and multiplying it by π – which he calculates but does not apply.*] [~EPE]

He constructs and measures a median from one triangle vertex. [*The use of the medians has no value in this problem and demonstrates his lack of understanding of the task.*] (Figure 11.) [~IPY, ~EMY] He calculates a ratio of the area of the triangle and the area of the circle. [~IMY, ~EPY] He attempts to apply a Boolean operator such that, if the measurement of the segment is less than the ratio of triangle area to circle area, an action should result. [~IMY, ~EPY] [*He flounders to determine what that resulting action should be.*]

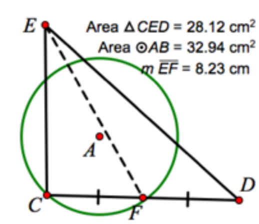

Figure 11

He begins attempting to make use of the coordinate grid and tries to make the points fall on exact lattice points to see how the triangle and circle interact. (Figure 12.) [~IPY, ~EPY] [*The background coordinate grid is seemingly hindering his problem solving. When elements are dragged around the screen, users are not guaranteed that these shapes will stay precisely on lattice points. (Note: There is a SNAP POINTS feature available to accomplish this, although, in this scenario, this would also have been of little value.)*]

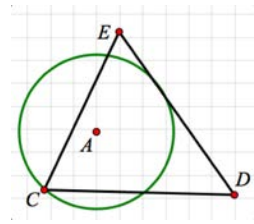

Figure 12

He begins to try to apply the triangle median as a heuristic, constructs the three medians, and experiments with Boolean operators for 15 minutes trying to discern what he wants to do next. (See previous Figure 9.) After realizing he has made no progress, he starts a new sketch, this time, with no grid in the background.

He constructs a disjoint circle and triangle. He highlights the triangle in its entirety and selects MARK CENTER. He sees no noticeable effect of this. [*MARK CENTER is intended to mark the center of a dilation or rotation. Neither of these operations would be meaningful at this point in the problem.*]

He [*again*] attempts to solve this problem using the triangle medians. (See previous Figure 9.) [~EPE] He selects all three medians and attempts to construct their intersection (the centroid). [*Sketchpad will construct the intersection of two elements, but not three simultaneous elements.*] He [*unsuccessfully*] looks for the feature for the intersection of all three and cannot find such. [*Not understanding this constraint within Sketchpad*], he gives up on attempting to find the intersection of the three medians and finds the intersection of two of the medians. He labels this point "triangle intersection." [*His technique of using the median will not be helpful and his label for this point is incorrect.*]

He measures the distance from the triangle centroid to the center of the circle. (See previous Figure 7.) [~IMY] [*Critical to this activity is determining and comparing distances.*] He continues working and [*believes he*] discovers the characteristics defining when the circle is completely inside the triangle. He uses a Boolean operator to report that, if the distance from the centroid to the center of the circle is less than the radius of the circle, the operator should report "1". However, the calculation reports "1" in all cases [*including incorrect cases*] [*The student seems to not have a global understanding of the technology and how it works.*]

He [*again*] briefly moves on to measuring and comparing the areas of the circle and triangle. [~EPY] [*This approach has little potential. He is incorrectly using the Boolean conditions and has an insufficient number of comparisons to use.*] [~ETY]

For many minutes, he measures all the sides of the triangle and just drags the sketch back and forth around the screen [*seemingly thinking of what to do next*].

He constructs two segments from the center of the circle to the intersections of the circle and triangle. (See previous Figure 4.) [~EMY] [*Not realizing this is always the length of the radius*], he tries to use this in comparison with Boolean commands. [~EMY, ~IMY] He tries to use these measurements as an “exist or not exist” feature. [~ETY] [*This will not work. More appropriately, he may want the Boolean condition to determine TRUE or FALSE*.] With only these two segments, he [*incorrectly*] creates Boolean conditions that compare the area of the circle and the triangle. [~EMY, ~ETY]

[*With time running out*], he finishes the activity without having any of the individual tasks completed.

These transcripts of student work involving problem solving in a DME demonstrate simultaneously that many activities and errors occur during the problem solving process. Thus, problem solving is a far more complex than simply performing a task.

Altogether, the preceding findings of domain, process, and interaction errors, dimensions within each of these, the Problem Solving Action Identity Framework, and the coding of transcripts lead to a number of implications. Some of these are provided in the following section.

Implications and Conclusion

A number of implications arise from the findings of this study. Some of these are discussed herein.

The transcripts above quite clearly demonstrate that the sequence of errors encountered in DME work is nonlinear and iterative. The nonlinearity of the sequence of errors connotes that errors are possibly less predictable than one would anticipate. A particular error in the DME problem solving process cannot predict the next error type. Additionally, the iterative nature of the sequence of error types reveals that a number of error types repeat after being separated by any number of other errors.

While the context of this investigation is student work in respect to DMEs, the findings of this study may possibly transcend the use of DMEs and be applicable to problem solving in general. In respect to the domains of knowledge associated with any problem solving activity, the problem solving actions (involving domains, processes, and interactions) and the accompanying error types (content, interpretation, activity, evaluation, syntactic, and semantic) may hold for any problem solving activity. Since the PAIF already recognizes problem solving as a domain of knowledge, the only modification that would be needed for different problem solving tasks would be to denote the other domains of knowledge involved. Employing the PAIF, it can be determined whether students struggle with the domain of a task or the task itself. This is invaluable in order to assess student understanding and misunderstandings within problem solving scenarios. It is hoped that future research involving other combinations of domains of knowledge and with or without the use of a technological infrastructure will lead to concretizing these results across problem solving in general.

The findings from this study imply that students generally need more training in the domain of problem solving to supplement instruction in other domains of knowledge in respect to the particular task at hand. Students need to both experience problem solving and study what are sound heuristic skills. However, in many cases, this may necessitate additional teacher training in the realm of problem solving.

This study has delineated a total of eighteen nonlinear and iterative error types associated with problem solving in a DME. As a consequence of this study and as fodder for future research, this study may have also captured all possible actions associated with problem solving processes in general. It is hoped that others can further this research to investigate these phenomena and validate or improve upon these findings.

References

Adu-Gyamfi, K. & Bossé, M. J. (2014). Processes and reasoning in representations of linear functions. *International Journal of Science and Mathematics Education*, *12*(1), 167-192.

Adu-Gyamfi, K., Bossé, M. J., & Chandler, K. (2015). Situating student errors: Linguistic-to-algebra translation errors. *International Journal for Mathematics Teaching and Learning*. Retrieved from http://www.cimt.org.uk/journal/bosse6.pdf.

Adu-Gyamfi, K., Stiff, L., & Bossé, M. J. (2012). Lost in translation: Examining translation errors associated with mathematics representations. *School Science and Mathematics, 112*(3), 159-170.

Ainsworth, S., & Van Labeke, N. (2004). Multiple forms of dynamic representation. *Learning and Instruction, 14(3)*, 241-255.

Arzarello, F., Micheletti, C., Olivero, F., Robutti, O., Paola, D., & Gallino, G. (1998). Dragging in Cabri and modalities of transition from conjectures to proofs in geometry. *Proceedings of the 22th annual conference of the International Group for the Psychology of Mathematics Education* (pp. 32-39). South Africa.

Arzarello, F., Olivero, F., Paola, D., & Robutti, O. (2002). A cognitive analysis of dragging practises in Cabri environments. *ZDM, 34*(3), 66-72.

Baccaglini-Frank, A. & Mariotti, M. A. (2010). Generating conjectures in dynamic geometry: The maintaining dragging model. *International Journal of Computers for Mathematical Learning, 14*, 225-253.

Bell, A., Brekke, G., & Swan, M. (1987). Misconceptions, conflict, and discussion in the teaching of graphical interpretation. Paper presented at the Second International Seminar in Misconceptions and Educational Strategies in Science and Mathematics, Ithaca, NY, July, 1987

Bogdan, R. C., & Biklen, S. K. (2003). *Qualitative research for education: An introduction to theories and methods* (4th ed.). Boston: Allyn and Bacon.

Bossé, M. J., Adu-Gyamfi, K., & Chandler, K. (2014). Students' differentiated translation processes. *International Journal for Mathematics Teaching and Learning*. Retrieved from http://www.cimt.org.uk/journal/bosse5.pdf.

Bossé, M. J., Adu-Gyamfi, K., & Cheetham, M. (2011a). Assessing the difficulty of mathematical translations: synthesizing the literature and novel findings. *International Electronic Journal of Mathematical Education*, *6*(3), 113-133.

Bossé, M. J., Adu-Gyamfi, K., & Cheetham, M. (2011b). Translations among mathematical representations: teacher beliefs and practices. *International Journal of Mathematics Teaching and Learning*. Retrieved from http://www.cimt.org.uk/journal/bosse4.pdf.

Carpenter, T. P. (1989). Teaching as problem solving. In R. I. Charles and E. A. Silver (Eds), *The teaching and assessing of mathematical problem solving* (pp.187-202). USA: National Council of Teachers of Mathematics.

Clement, J., Lockhead, J., & Monk, G. S. (1981). Translation difficulties in learning mathematics. *American Mathematical Monthly, 88*, 287-290.

Cobb, P., Wood, T. and Yackel, E. (1991). A constructivist approach to second grade mathematics. In von Glaserfield, E. (Ed.), *Radical constructivism in mathematics* *education* (pp. 157-176). Dordrecht, The Netherlands: Kluwer Academic Publishers.

Cobb, P., Yackel, E., & Wood, T. (1992). A constructivist alternative to the representational view of mind in mathematics education. *Journal for Research in Mathematics Education*, *23*(1), 2-33.

Creswell, W. J. (2003). *Research design: Qualitative, quantitative and mixed methods approaches* (2nd ed.). London: Sage Publications.

DePeau, E. A. & Kalder, R. S. (2010). Using dynamic technology to present concepts through multiple representations. *Mathematics Teacher, 104*(4), 268-273.

Dick, T. P., & Hollebrands, K. F. (2011). Introduction to *Focus in high school mathematics: Technology to support reasoning and sense making*. In T. P. Dick & K. F. Hollebrands (Eds.), *Focus in high school mathematics: Technology to support reasoning and sense making* (pp. xi - xvii). Reston, VA: National Council of Teachers of Mathematics.

Dreyfus, T., & Eisenberg, T. (1987). On the deep structure of functions. *Proceedings of the eleventh international conference of the International Group for the Psychology of Mathematics Education* (Vol. l, pp. 190-196). Montreal: IGPME.

Dunham, P., H., & Osborne, A. (1991). Learning how to see: Students' graphing difficulties. *Focus on Learning Problems in Mathematics*, *13*, 35–49.

Duval, R. (1999). Representation, vision and visualization: Cognitive functions in mathematical thinking: Basic issues for learning. In F. Hitt & M. Santos (Eds.), *Proceedings of the twenty first annual meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education*, 3-26.

Duval, R. (2006). A cognitive analysis of problems of comprehension in the learning of mathematics. *Educational Studies in Mathematics, 61*(1-2), 103-131.

Edwards, Michael T. (2003). Visualizing transformations: matrices, handheld graphing calculators, and computer algebra systems. *The Mathematics Teacher*, *96*(1), 48-56.

Evan, R. and Lappin, G. (1994). Constructing meaningful understanding of mathematics content. In Aichele, D. and Coxford, A. (Eds.) *Professional Development for Teachers of Mathematics* (pp. 128-143). Reston, Virginia: NCTM.

Finzer, W. & Bennett, D. (1995). Technology tips: From drawing to construction with The Geometer's Sketchpad. *Mathematics Teacher, 88*(5), 428-431.

Gagatsis, A., & Shiakalli, M. (2004). Ability to translate from one representation of the concept of function to another and mathematical problem solving. *Educational Psychology, 24*(5), 645-657.

Garofalo, J., Drier, H., Harper, S., Timmerman, M.A., & Shockey, T. (2000). Promoting appropriate uses of technology in mathematics teacher preparation. *Contemporary Issues in Technology and Teacher Education* [*Online serial*], *1*(1). Available: http://www.citejournal.org/vol1/iss1/currentissues/mathematics/article1.htm

Goldenberg, E. Paul, and Albert A. Cuoco. 1998. "What is dynamic geometry?" In R. Lehrer & D. Chazan (Eds.), *Designing learning environments for developing understanding of geometry and space* (pp. 351-367). Mahwah, NJ: Lawrence Erlbaum Associates.

Goldin, G. A. (2000). A scientific perspective on structure: task-based interviews in mathematics education research. In A. Kelly & R. Lesh (Eds.), *Handbook of research design in mathematics and science education* (pp. 517-546). Mahwah, NJ: Lawrence Erlbaum Associates.

Graham, C. R., Borup, J., & Smith, N. B. (2012). Using TPACK as a framework to understand teacher candidates' technology integration decisions. *Journal of Computer Assisted*

Learning, 28(6), 530-546.

Hollebrands, K. (2003). High school students' understandings of geometric transformations in the context of a technological environment. *Journal of Mathematical Behavior, 22*(1), 55-72.

Hollebrands, K. F., & Lee, H. S. (2012). *Preparing to teach mathematics with technology: An integrated approach to geometry*. Dubuque, IA: Kendall Hunt.

Hoyles, C., & Noss, R. (1994). Dynamic geometry environments: What's the point? *Mathematics Teacher, 87*(9), 716-717.

Janvier, C. (1987). Translation process in mathematics education. In C. Janvier (Ed.), *Problems of representation in mathematics learning and problem solving* (pp. 27-31). Hillsdale, NJ: Lawrence Erlbaum Associates.

Kaput, J. J. (1987a). Representation systems and mathematics. In C. Janvier (Ed.), *Problems of representation in teaching and learning mathematics* (pp. 19–26). Hillsdale, NJ: Erlbaum.

Kaput, J. J. (1987b). Toward a theory of symbol use in mathematics. In C. Janvier (Ed.), *Problems of representation in mathematics learning and problem solving* (pp. 159-195). Hillsdale, NJ: Erlbaum.

Kerslake, D. (1981) Graphs. In K. Hart (Ed.), *Children's understanding of mathematics: 11-16* (pp. 120-136). Oxford: John Murray.

Knuth, E. J. (2000). Student understanding of the Cartesian connection: An exploratory study. *Journal for Research in Mathematics Education*, *31*(4).

Koehler, M. J., & Mishra, P. (2005). What happens when teachers design educational technology? The development of technological pedagogical content knowledge. *Journal of Educational Computing Research, 32*(2), 131–152.

Lee, H. S., Harper, S., Driskell, S. O., Kersaint, G., & Leatham, K. (2012). Teachers' statistical problem solving with dynamic technology: Research results across multiple institutions. *Contemporary Issues in Technology and Teacher Education [Online serial], 12*(3). http://www.citejournal.org/vol12/iss3/mathematics/article1.cfm.

Lee, H. S., & Hollebrands, K. F. (2006). Students' use of technological features while solving a mathematics problem. *Journal of Mathematical Behavior, 25*, 252-266.

Leinhardt, G., Zaslavsky, O., & Stein, K. M. (1990). Functions, graphs and graphing: Tasks, learning and teaching. *Review of Educational Research, 60*(1), 1-64.

Lester, F. K. Jr., Masingila, J. O., Mau, S. T., Lambdin, D. V., dos Santon, V. M. and Raymond, A. M. (1994). Learning how to teach via problem solving. In Aichele, D. and Coxford, A. (Eds.), *Professional Development for Teachers of Mathematics* (pp. 152-166). Reston, Virginia: NCTM.

Lopez-Real, F., & Leung, A. (2006). Dragging as a conceptual tool in dynamic geometry environments. *International Journal of Mathematical Education in Science & Technology*, *37*(6), 665-679.

Miles, M. B. & Huberman, M. N. (1994). *Qualitative data analysis: an expanded sourcebook.* Thousand Oaks, CA: Sage.

Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, *108*(6), 1017–1054.

National Council of Teachers of Mathematics (NCTM). (2000). *Principles and standards for school mathematics*. Reston, VA: Author.

Niess, M. L. (2005). Preparing teachers to teach science and mathematics with technology:

Developing a technology pedagogical content knowledge. *Teaching and Teacher Education*, *21*(5), 509–523.

Niess, M. L. (2006). Guest editorial: Preparing teachers to teach mathematics with technology. *Contemporary Issues in Technology and Teacher Education*, *6*(2), 195–203.

Olkin, I. & Schoenfeld, A. (1994). A discussion of Bruce Reznick's chapter. In A. Schoenfeld (Ed.), *Mathematical Thinking and Problem Solving* (pp. 39-51). Hillsdale, NJ: Lawrence Erlbaum Associates.

Özgün-Koca, S. A., Meagher, M., & Edwards, M. T. (2010). Preservice teachers' emerging TPACK in technology-rich methods class. *Mathematics Educator, 19*(2), 10-20.

Pea, Roy D. 1985. Beyond amplification: Using the computer to reorganize mental functioning. *Educational Psychologist*, *20*(4): 167–182

Preece, J. (1983). Graphs are not straightforward. In T. R. G. Green, & S. J. Payne (Eds.), *The psychology of computer use: A European perspective* (pp. 41–50). Academic Press, London.

Ruthven, K., Hennessy, S., & Deaney, R. (2008). Constructions of dynamic geometry: A study of the interpretive flexibility of educational software in classroom practice. *Computers & Education 51*(1): 297-317.

Schifter, D. and Fosnot, C. (1993). *Reconstructing Mathematics Education*. NY: Teachers College Press.

Schmidt, D. A., Baran, E., & Thompson, A. D. (2009). Technological pedagogical content knowledge (TPACK): The development and validation of an assessment instrument for preservice teachers. *Journal of Research on Technology in Education*, *42*(2), 123-149.

Schoenfeld, A. (1994). Reflections on doing and teaching mathematics. In A. Schoenfeld (Ed.), *Mathematical Thinking and Problem Solving* (pp. 53-69). Hillsdale, NJ: Lawrence Erlbaum Associates.

Stacey, K. and Groves, S. (1985). *Strategies for problem solving*. Melbourne, Victoria: VICTRACC.

Stanic, G. and Kilpatrick, J. (1989). Historical perspectives on problem solving in the mathematics curriculum. In R. I. Charles and E. A. Silver (Eds.), *The teaching and assessing of mathematical problem solving* (pp.1-22). USA: National Council of Teachers of Mathematics.

Strauss, A., & Corbin, J. (1990). *Basics of qualitative research*. London: Sage Publications Ltd.

Thompson, P. W. (1985). Experience, problem solving, and learning mathematics: considerations in developing mathematics curricula. In E. A. Silver (Ed.), *Teaching and learning mathematical problem solving: Multiple research perspectives* (pp.189-236). Hillsdale, N.J: Lawrence Erlbaum.

Van Zoest, L., Jones, G., & Thornton, C. (1994). Beliefs about mathematics teaching held by pre-service teachers involved in a first grade mentorship program. *Mathematics Education Research Journal*. *6*(1): 37-55

Zelkowski, J., Gleason, J., Cox, D. C., & Bismark, S. (2013). Developing and validating a reliable TPACK instrument for secondary mathematics preservice teachers. *Journal of Research on Technology in Education*, *46*(2), 173–206.

Zhang, J. (1997). The nature of external representations in problem solving. *Cognitive Science*, *21*(2), 179-217.

Author Information

Molly Brown is a teacher of secondary mathematics at West Lincoln High School in Lincolnton, NC. She has recently earned her Masters Degree in Secondary Mathematics Education. She anticipates continuing to enhance her teaching and learning through various research opportunities.

Michael. J. Bossé is the Distinguished Professor of Mathematics Education and MELT Program Director at Appalachian State University, Boone, NC. He teaches undergraduate and graduate courses and is active in providing professional development to teachers in North Carolina and around the nation. His research focuses on learning, cognition, and curriculum in K-16 mathematics.

Kayla Chandler is a mathematics education doctoral candidate at North Carolina State University. She has experience teaching both high school mathematics and collegiate mathematics education courses. She currently serves as a graduate research assistant for the Preparing to Teach Mathematics with Technology (PTMT) project at North Carolina State University and as an adjunct faculty member for East Carolina University. Her research interests include teacher noticing of students' mathematical thinking and preparing teachers to successfully implement and utilize technology in the mathematics classroom to enhance student learning.