The Development of a Mathematics Task Coding Instrument (MaTCI)

M. Kathleen Heid
The Pennsylvania State University

Glen Blume
The Pennsylvania State University

Karen Hollebrands
North Carolina State University

Cynthia Piez
University of Idaho

Paper presented at the Research Presession at the 80th Annual Meeting of the National Council of Teachers of Mathematics, Las Vegas, Nevada, April 21, 2002

This research was supported by the National Science Foundation under Grant No. TPE 96-18029 to The Pennsylvania State University with major subcontract to The University of Iowa. Additional funding for the implementation component of this research project came from the Northern Kentucky University Center for Integrative Science and Mathematics (CINSAM). Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily reflect the views of the National Science Foundation or of Northern Kentucky University.

A defining characteristic of today's mathematics classrooms is that, in them, students engage in performing mathematical tasks. The National Council of Teachers of Mathematics (1991) has proclaimed the importance of "worthwhile mathematical tasks," and researchers have sought explanations for their impact on student learning. The CAS-IM curriculum modules are integrated technology-intensive mathematics modules designed to engage students in mathematical tasks that would develop their understandings of "big ideas" in mathematics like transformation, function, equivalence, and iteration. Particularly important in the design of these materials was that students would use the computer as a tool for exploring an organized set of facets of these "big ideas." The written curriculum, however, is almost nowhere equivalent to the experienced curriculum. As teachers and students use curriculum materials, there is invariably a conscious or unconscious mediation of students' lived experience of the curriculum. In a technology-intensive curriculum, the technology serves as a further mediator of the experience. Of particular interest in this study was how students and teachers used, sequenced, and otherwise dealt with curriculum tasks related to those in the CAS-IM curriculum.

For at least the past several decades, researchers have been studying the role of academic tasks in instruction (Doyle, 1983, 1988; Doyle & Carter, 1984; Nicely, 1970), and, more recently, their investigations have turned to the role of mathematical tasks in reform-oriented mathematics instruction (Stein, Grover, & Henningsen, 1996). These researchers have categorized mathematics teachers' classroom questions (Friedman, 1974, 1976), textbook tasks (Nicely, 1970, 1985), and both textbook tasks and teacher questions (Engelder, 1991). These earlier category systems were reminiscent of Bloom's taxonomy, and, although applied to mathematics classes and texts, used categories (e.g., apply, relate, convert, symbolize, summarize, describe) that could have also served well across a range of other subject matter areas. Later category systems like that used by Stein and her colleagues (Stein, Grover, & Henningsen, 1996) based themselves in mathematical activity, but did not need to and their categories were not designed to distinguish among higher levels of mathematical activity (levels that required more complex thinking or problem solving). The categories used by Stein and her colleagues were: memorization, procedures without connections, procedures with connections, and doing mathematics. Kawanaka and Stigler (1999) grouped the cognitive demands of tasks into three categories: those requiring "yes/no" responses, those requiring students to name or state a short response, and those requiring students to describe, explain or reason. Although they applied their coding schemes to mathematics classrooms, difficulty in achieving interrater reliability resulted in abandoning a plan to "capture differential mathematical communication and cognitive processes … such as justification or analysis." (p. 257) Like the categories used by Stein and her colleagues, those used by Kawanaka and Stigler were not designed to capture the nuances of non-routine mathematical tasks. Doyle (1988) defined a task as calling attention to four aspects of work in a class describable as the desired goal state or end product, the problem conditions and resources available, the operations involved in using the resources, and the overall importance of the task. Our work focused primarily on the goal state or end product of the activity.

Because there has been such a history of research on the nature of mathematical tasks and questions as they play out in the mathematics classroom, one might wonder why we took on the task of creating a new category system for mathematical tasks. The rationale for our development of this new category system derives from the context for our study. We were examining students' mathematical activity in the context of a technology-intensive "reformed" high school mathematics curriculum. It was our assumption that availability of powerful and familiar technological tools (dynamic tools like Geometer's Sketchpad and computer algebra systems like the CAS on the TI-92) would foster higher level mathematical activity, in particular work in the areas of conceptual reasoning and justification. Because students would be familiar with the tools, they would not hesitate to use them, and because the tools performed the complete gamut of routine operations, students could focus on more conceptual ideas. In addition, reform curricula like ours were designed to engage students in thinking about overarching mathematical ideas with a focus on conceptual thinking instead of on execution of procedures. Because the new curriculum we were using made intensive use of technology to develop overarching mathematical ideas, the coding systems developed by the aforementioned researchers were not suited to our needs. We needed to develop an instrument that would allow us to capture the sequence of tasks students took on as they worked with technology on conceptually oriented tasks.

The Coding Instrument

Early in the analysis process, we decided it would be fruitful to classify the tasks on which students were working, and, as previously mentioned, existing instruments were not suitable for our purposes either because of their generic nature or because they did not allow for discriminating among any tasks except those at primarily routine levels. We chose to focus on, in terms of Doyle's scheme, the desired goal state or end product. To describe the tasks on which the students were working, we felt that it would not be sufficient to categorize the tasks presented in the curriculum on which they were working. With the close involvement of the researchers and with the collection of data within regularly scheduled classes, there was bound to be interference with the tasks as posed. Feeling the need to broaden the definition of task beyond the set of mathematical activities posed in the curriculum materials, we needed to decide on what constituted a task. Our concept of “task” involved the following assumptions:

- Tasks are goal-driven and require student action and “new” thinking. Assumes students are not just reporting previously recorded findings.
- Tasks that to be coded are those that are posed by the researcher, curriculum, teacher, and students. Tasks may be implicit or explicit.
- While students are working on a large task, such as the one posed by the curriculum, subtasks may emerge [tasks will overlap].

We began (the process was iterative) by deciding to code the tasks on which students seemed to be working regardless of the initial task or its initiator rather than solely those that were posed by the curriculum. This expanded the set of tasks to include the curriculum tasks, tasks as they were interpreted by the students, tasks posed by the researcher or teacher, and tasks posed by the students. Consequently, the mathematics task coding instrument (MaTCI) was developed to document the types of mathematical tasks on which students were working throughout the study’s small group sessions.

Creation of codes. Creation of categories and codes was grounded in the transcript data. The first step was to generate a potential list of types of tasks from a sample of the transcripts. Working from annotated transcripts of a subset of the class sessions during which the small group work occurred, several of the researchers identified the tasks on which students were working and categorized the types of task by the cognitive mathematical work required. Codes like “describe observation” and “evaluate equation” resulted. The resulting set of categories and codes was piloted by using it to categorize the tasks identified in transcripts of other class sessions. Categories and definitions were revised based on these trials. Little used categories were eliminated, and categories that were difficult to distinguish were collapsed. This process of code-define-recode-redefine-recode was carried out iteratively. The final task codes are shown in the table following this paragraph. In creating these codes, we paid attention to what the action was (e.g., describe, produce, corroborate) and what was being acted on (e.g., input or output value, result, phenomenon, procedure, generalization). To facilitate and organize the coding, we sorted the coding tasks into three categories: Concept: tasks whose aim was the characterization of a concept; Product: tasks whose aim was the generation of a particular mathematical object; and Reasoning: tasks whose aim was to provide a complete or partial rationale for a conclusion.

Category	Subcategory	Code
Concept	identify	identify object
	describe	observation or procedure
	elaborate	compare/explain/describe phenomenon
Product	produce	Produce a value or an output given an input value; input value given an output value; Produce a graph
	generate	function specifics; a procedure
	predict	predict
	generalize	generalize
Reasoning	corroborate	a procedure or a generalization
	justify	justify

When the current research team reached agreement on the categories and definitions, members of the larger research team were trained to code transcripts and the revision process was again carried out. Each time, the definitions were sharpened and the categories and codes were refined. Throughout the development of the coding system, several questions were of recurring concern:

What constitutes a task?

How should we treat the notion of overlapping tasks?

How small (or large) could a task be?

How should we differentiate between justification and corroboration?

How should we differentiate between CED and justification?As we broadened our group of coders, these questions arose from a number of the coders less familiar with the curriculum or our research.

Levels of coding. We have grouped the types of tasks into three overarching categories, Concept, Product, and Reasoning. The subcategories with each category seemed to reflect different cognitive levels related to the category. [We have not performed any hierarchical analysis on task level, so we do not claim anything but apparent (to us) cognitive level.] Within the Concept category, we believe that the order of cognitive difficulty (from lowest to highest) is “Identify,” “Describe,” and “Elaborate.” Within the Product category, we believe that the order of cognitive difficulty (from lowest to highest) is “Produce,” “Generate,” “Predict,’ and “Generalize.” Within the Reasoning category, we believe that the order of cognitive difficulty (from lowest to highest) is "Corroborate" and "Justify."

Coding of sample transcript. The four researcher/authors conducted the coding on two two-day small group sessions. Two researchers coded each transcript separately, comparing task chunks, categories, and codes, and coming to agreement on differences. When discussion did not resolve the differences, the four-person team revised the coding to accommodate the differences. Table 1 gives the coding categories we used for this report, Appendix 1 contains a partial transcript of students working on the problem shown in Figure 1, and Appendix 2 shows the Task codes for the partial transcript.

Connection IIa: Ceiling Function Family and Inverses

1. In section one you examined function families that included linear, quadratic, rational, and exponential families. In the following situation there is a relationship between two variables. Read the situation in part a and then answer Items i. – v.

   a. An employer pays each employee a base salary of $24 dollars a day plus an additional $4.10 per hour.

      i. Write a rule that accepts as input the number of hours worked and gives as output the earnings per day. Explain why your rule makes sense in terms of the situation.

      ii. As an employee you may want to determine how many hours you need to work in a day to earn a specific amount of money. Write a rule that accepts as input the earnings per day and gives as output the number of hours worked. Explain why your rule makes sense in terms of the situation.

      iii. Explain how the rule in part i is related to the rule in part ii.

Figure 1. Problem on which students were working in partial transcript

Coding using MaTCI provides opportunities for several types of analysis. First, the instrument features the display of overlapping tasks. We were coding the task on which students seemed to be working at a given time. At the same time, however, there were what we came to call " lurking tasks," tasks that had been posed but that had taken a temporary back seat to the current attention students and researcher were placing on what we called “active tasks.” The coding of the partial transcript (Appendix 1) shown in Appendix 2 illustrates a number of features of the task sequencing that can help to elucidate the sequencing of tasks as they play out in a technology-rich curriculum. The partial transcript codes identify the type of task, the initiator of the task, and the line numbers reflecting the part of the class during which that task was “on the table.” Tasks were considered to be “on the table” under two different circumstances. First, it could be that students were actively working on the task. In this case, we considered the task an “active task,” and usually recorded it in the column labeled Task 1 or Task 2 (See Appendix 2). Second, the task could have been one that was once active but was put on the side while a subtask or related task was being worked on. In this case, we considered the task a “lurking task.” This was usually a curriculum-initiated task and was recorded in the “Curriculum” column (See Appendix 2).

The curriculum column shows the sequence of tasks that ostensibly drives the class session. These are usually tasks posed in the curriculum materials. A chart like the one in Appendix 2 illustrates when several tasks may be under consideration and in what order. The chart also shows the sequence and nature of tasks proposed in the curriculum or by the teacher, the researcher, or the students. One can locate intervals in the transcript when the answer to a higher level task, for example, is generated through the execution of a series of lower-level tasks. One can study the interaction between active tasks and lurking tasks of different types. In a separate paper, we show how we have used the task codes to examine the interplay of tasks.

Level 1 Category	Level 2 Category	Code	Explanation	Example
Concept	*Identify*	Identify object (IO)	The task is to identify the *name* of an object when the characteristics of that object are presented to the student.	What do you call a triangle that has one right angle?
	Describe	Describe observation (DO)	The task is to state what you see (visual to perception) describing/characterizing a mathematical object.	The slidergraph jumps when the parameter changes from positive to negative and the researcher asks, "Can you tell me what you see happening?"
		Describe procedure (DP)	The task is to describe or identify a procedure that is already known or observable by the student.	The researcher asks, "what does the floor of x do to a number?"
	Elaborate	Compare/Explain/Describe phenomenon (CED)	Several different types of tasks fit under this category: Type 1: The task is to compare two different mathematical objects (e.g., functions, expressions) or representations (e.g., graphs, tables, symbols, rules, numerical values). Although comparison is often involved in explanation and justification the focus is on finding connections between two or more phenomena each of which is of interest. Type 2: The task is to make sense of one thing in terms of something else. Type 3: The task is to describe or identify a phenomenon that is not purely perceivable by the senses or to describe or identify a procedure.	Type 1: A student suggests using the TI-92 calculator to graph the ceiling and floor functions on the same grid so they can see them together. Type 2: The students graph the ceiling function on the TI-92 and the graph that appears is connected in a stair-step fashion. The researcher asks, "Does this make sense in terms of what you know about the ceiling function?" Type 3: The researcher asks, "what is an inverse function?"

Product	*Produce*	Produce a value or output given an input (Evaluate) PE	The task is to provide one or a set of output value(s) given one (or a set of) particular input value(s).	The teacher asks the students, “What is the floor of negative ten point six?”
		Produce input given output (Solve) PS	The task is to provide one or a set of input value(s) given one (or a set of) particular output value(s).	The students are working on the task of determining the values of A, B, C, and D so that A floor(B(x-C))+D= 0
		Produce a graph PG		
	Generate	Generate function specifics GFS	The task is to produce a function rule. Given a rule, graph, table, or situation.	The curriculum provides students with a graph of a function and they need to produce a function rule.
		Generate a procedure GP	The task is to create a procedure that does not already exist.	The task for the students is to describe or generate a new procedure. For example, describe what to do to one column to find the values in the other column.
	Predict	Predict (Pt)	The task is to describe what might happen under certain conditions in a novel situation. Students are asked to come up with a conjecture. There is no expectation that the students have enough information to deduce the answer.	The researcher asks, “If you changed the value of D, which is what you’re working on right now, what would you expect to happen to the graph?”
	Generalize	Generalize (G)	*Generate* a relationship that holds for an entire class. The task is to generate a relationship from instances that are given or from logic.	The task posed on the handout from which students are working asks, “What are the effects of changing the value of D?”

Reasoning	*Corroborate*	Corroborate (disconfirm or check) finding. result/ answer (CR) procedure (CP) or generalization (CG)	The task is to provide additional evidence that what is given or found is true. [Determine if something might be true or false] *There must be a suspected-to-be-true item in the air.*	A student finds the output values for a given input value for the floor function using his head. Another student uses the calculator to check the student's answer.
	Justify	Justify (J)	The task is to provide a logical argument for why something happens [Establish truth of finding *in the spirit of the field of mathematics rather than in terms of what convinces the student.*]	A student notices that changing the value of D causes the graph to move vertically. The researcher asks, "Why do you think it moves vertical?"

Table 1. Category codes and categories for the MaTCI instrument.

Conclusion

We anticipate that the MaTCI will be useful in analyzing student work on tasks when those tasks are more fine-grained and fast-paced than the ones in studies like those of Stein and her colleagues or of Kawanaka and Stigler. Stein and colleagues reported that their tasks were typically 24 minutes long, while some of our tasks are less than a minute long. Other studies have focused on one task at a time, while our tool opens the door to analyzing more than one simultaneous task. We expect that, when combined the Task analysis is coordinated with analysis based on the Tech categories and the MAGICAL Representation categories, the result will be a rich understanding of the interplay of tasks, representations and technology in the context of reform-oriented curricula. We expect that as we analyze data from the CAS-IM project, we will continue to refine the MaTCI.

References

Doyle, W. (1988). Work in mathematics classes: The context of studnets' thinking during instruction. *Educational Psychologist* 23(2): 167-180.

Doyle, W. & Carter, K. (1984). Academic tasks in classrooms. *Curriculum Inquiry*, 14, 129-149.

Engelder, J. G. W. (1991). A comparison of thinking skills elicited in textbooks used in computer-intensive algebra and traditional algebra. Unpublished masters paper, The Pennsylvania State University.

Friedman, M. (December, 1974). A system to analyze geometry teachers' questions. *Mathematics Teacher*, 709-713.

Friedman, M. (November, 1976). Cognitive emphasis of geometry teachers' questions. *Journal for Research in Mathematics Education*, 259-263.

Kawanaka, T. & J. W. Stigler (1999). Teachers' use of questions in eighth-grade mathematics classrooms in Germany, Japan, and the United States. *Mathematical Thinking and Learning* 1(4): 255-278.

Nicely, Jr., R. F. (April, 1985). Higher-order thinking skills in mathematics textbooks. *Educational Leadership*, 26-30.

Nicely, Jr., R. F. (1970). Development of procedures for analyzing materials for instruction in complex numbers at the secondary level. Unpublished Ph. D. dissertation, University of Pittsburgh.

Stein, M. K., B. W. Grover, and M. Henningsen (1996). Building student capacity for mathematical thinking and reasoning: An analysis of mathematical tasks used in reform classrooms. *American Educational Research Journal* 33(2): 455-488.

Appendix 1: Partial Transcript

Appendix 2: Sample Coding of Partial Transcript