Coder's Manual

Codebook for Categorization of Highway Formula Protocols

Robert M. Hamm
Center for Research on Judgment and Policy
Institute of Cognitive Science
University of Colorado

Campus Box 344
Boulder, CO 80309-0344

Technical Report No. 88-8

September 1985

revised December 1987

This work was sponnsored by the Engineering Psychology Program, Office of Naval Research, Contract N00014-81-C-0591, with additional support by the U.S. Amy Research Institute for the Behavioral and Social Sciences, Contract MDS-903-86-K-0265.

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION	
1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
3. DISTRIBUTION/AVAILABILITY OF REPORT	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	
5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION	Institute of Cognitive Science
6b. OFFICE SYMBOL (If applicable)	
7a. NAME OF MONITORING ORGANIZATION	
6c. ADDRESS (City, State and ZIP Code)	University of Colorado, Box 345 Boulder CO 80309-0345
7b. ADDRESS (City, State and ZIP Code)	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	Army Research Institute
8b. OFFICE SYMBOL (If applicable)	
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	MDA-903-86-K-0265
8c. ADDRESS (City, State and ZIP Code)	5001 Eisenhower Avenue Alexandria VA 22333-5600
10. SOURCE OF FUNDING NOS.	PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
11. TITLE (Include Security Classification)	Codebook for Categorization of Highway Formula Protocols.
12. PERSONAL AUTHOR(S)	Robert M. Hamm
13a. TYPE OF REPORT	Technical
13b. TIME COVERED	FROM ____ TO ____
14. DATE OF REPORT (Yr., Mo., Day)	1988 March
15. PAGE COUNT	
16. SUPPLEMENTARY NOTATION	Contracting Officer's Representative was Michael Drillings.

17. COSATI CODES		
FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Analysis/Intuition, Problem Solving, Experts, Engineers, Highway judgments, Protocol analysis, Use of mathematical knowledge.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This document is the Coder's Manual for the protocol analysis used to measure "Moment by Moment Variation in Experts' Analytic and Intuitive Cognitive Activity" (Hamm, Robert; Institute of Cognitive Science and Center for Research on Judgment and Policy, University of Colorado, Boulder, 1985). The study analyzed the thinking-aloud verbalizations of highway engineers, who were asked to produce a mathematical formula that expresses the dependence of highway safety, capacity, or aesthetic value, on a number of appropriate cues. The Coder's Manual defines every category in each of fourteen categorization schemes, and reports the reliability with which each category and each scheme was coded.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS ☐
21. ABSTRACT SECURITY CLASSIFICATION	
22a. NAME OF RESPONSIBLE INDIVIDUAL	
22b. TELEPHONE NUMBER (Include Area Code)	
22c. OFFICE SYMBOL	

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

SECURITY CLASSIFICATION OF THIS PAGE

TABLE OF CONTENTS

Introduction

This paper contains the instructions used in analyzing the transcripts of highway engineers producing formulas expressing the aesthetic value, safety, or capacity of highways, as reported in Hamm (1985; 1987) and in Hammond, Hamm, Grassia, and Pearson (1987). The transcripts are segmented and each is coded with respect to nineteen categorization schemes. Fourteen of these schemes (A, B, C, D, E, G, H, J(Pa, Re, Ch, Inc, Mut), K(Cnf, Dbt)) address the extent to which the segment involves intuitive or analytical cognition, three (I, II, F) are concerned with the type of reasoning task the subject is defining for him or her self, one (III) addresses the kind of information processing the subject is doing, and the last (IV) records whether the subject is reporting his or her cognition concurrently or retrospectively.

Applicability to Other Tasks

Although this codebook explains the application of these categorization schemes to highway engineers' formula making, most of the concepts are applicable to other kinds of problem solving, judgment, or decision making tasks. The usefulness of individual categorization schemes will depend on the subject's task and on the researcher's hypotheses. See Hamm (1987) for the evaluation of these categorization schemes as applied in measuring analytical and intuitive activity by highway engineers when producing formulas.

Sources of the Categorization Schemes

The categories were selected by the author in consultation with colleagues and with guidance from the sources cited below. The codebook was used to guide the coding process, and clarifications were added as issues arose.

Categorization Schemes

For easy reference a synopsis of the categorization schemes, or levels, is given in Appendices A, B, C, and D.

Reliability of the Categorization Schemes

The reliability of each category in each level and of each level is discussed in the last section of this manual.

Segmentation of Transcripts

What size of unit should be used in categorization? Although some units of cognition in these transcripts are several sentences long (even paragraphs, e.g., in reading the information sheets), others are less than a sentence. To handle this variety in the size of the units of cognition, we segment the transcripts conservatively, that is, into pieces that are too small rather than too large. The segments are delineated according to very simple rules. Long sentences are broken up into clauses, particularly if there are separate judgments or topics in each. The coders are instructed to pay attention to surrounding clauses in judging what is going on in each one. Hence, it is better to break up a sentence too much rather than too little. However, there is also the danger of breaking up something too much, and hence losing sight of the big thing that is happening.

Procedure: One copy is made of each transcript, and the segment boundaries are marked and numbered sequentially. Coding sheets on which the rows are numbered sequentially, for the segments, and the columns represent the categorization schemes, or levels, are used. The coder will assign only one category to each segment at each level.

Gaps

If there is an obvious gap, where the engineer "must have" done something but we do not have evidence for it in the transcript, we do not have to code it. Examples: he must have thought about the organizing principle, because now he is using one; he must have generated a plan, because now he is following it; he must have remembered that idea, because it is not in our information packet and he could not have just judged it; but there is no sentence or unit that can be coded to reflect this! Since we are not simulating, we do not need to identify these "necessary steps".

General Orientation toward Encoding These Cognitive Continuum Categories

The size of the unit that we will encode will vary from Level to Level. At some of the Levels, whole sentences, or even sequences of sentences, can be encoded with the same concept, and all of the numbered segments given the same codes. At other Levels, coding decisions will need to be made for each segment.

The engineers' reading of the instructions should not be coded; nor should their reports of their formulas.

In most cases, tangential remarks in the middle of long sentences should be coded appropriately, or even not coded at a particular level, rather than being given the same code as the surrounding segments.

Making Use of Context

Coders are instructed to make use of context in interpreting a segment. This is necessary given how small we chose to make the segments, and given other difficulties in interpreting isolated statements. However, it means that each of the segments is not necessarily independent of its surrounding segments.

Categorization Schemes Pertaining to
the Degree of Intuition or Analysis

Codes for Decisions and Justifications

Many of the steps of building a formula can be considered "actions" that the person takes. Thus, there is an explicit or implicit decision behind each one. We have two different sets of categories for addressing this kind of activity.

Level A: Decisions and Decision Making

Use this Level when it is clear that the engineer is taking an action, making a decision, or considering one hypothetically in an "if ... then ..." construction. At this level, we should only code actions that have to do with the building of a formula, including

1. Information steps - deciding what to look at, what to pay attention to

2. Picking a goal or a method of attaining a goal

3. Establishing constraints

4. Generating formulas or formula parts

5. Doing something to get a different perspective, e.g., as part of an evaluation

6. Stating an evaluation of part of one's formula, or an evaluation of a strategy for producing such a part

If there is a long discussion of a decision, one or another code from this level should be applied to each segment in the discussion. These categories are listed here roughly in order of increasing analyticity.

1. Act - The engineer simply takes an action, without verbally acknowledging in this segment that he is making a decision. He describes the action he is taking, without saying "I guess I will ..." or "Then I'll ..." and without expressing any sense of hesitancy. If he gives a reason or acknowledges he is making the decision in another segment, but does not do so in this one, then this one should be coded as "Act".

2. Cs - Conscious. The engineer acknowledges that he is making a decision here (or trying to). Thus, it is apparent that he is holding it up for his own inspection. Perhaps he expresses hesitancy. Yet there is no presentation in this segment of reasons or justifications for his decision.

   The point of the distinction between this category and mere "action" is that the engineer is clearly "making a decision" instead of just "taking an action". He is thinking about what to do here. Since we do not have evidence in this segment of his thinking, that is, his reasons, comparisons, etc., our only evidence for this thinking is his verbal or nonverbal reference to it. Thus, if an act has one of the social markers, e.g., "OK ..." or "All right ...", this suggests it is a conscious decision [unless there is a big pause between the marker and the action]. However, if you see that a segment marked by an "OK" is just a mindless act, a reflex, a following of a plan, then it should be marked "Act" rather than "Cs".

Planning, using the future tense, also suggests a "Cs" kind of activity. Another example is the justification of an action by reference only to one's own authority, e.g., "I think I'll do this".

3. Opt - Describing an option. If the engineer makes a long description of an option (whether concrete or hypothetical), it should be coded as an action only at the moment at which he clearly does it, or decides to do it; other segments of this long option-description should be coded "Opt" unless they have elements of comparison (Com), reasons (Rea), tradeoffs among multiple attributes (Tra), probabilities and contingencies (Dt), or expected values (For).

4. Ev - If the engineer evaluates (positively or negatively) a proposed action or a previously made decision, contradicts it, approves it, questions it, takes it back (hence is aware that he is making the decision), yet does not give explicit justification in this segment for having changed his mind. Prototype: "No, that's no good."

5. Com - The engineer compares one possible action with another, in a minimal sense, without giving reasons (Rea) or discussing tradeoffs (Tra). Prototype: "It is better to do it this way than that way."

6. Rea - The engineer gives reasons or justifications for his decision, using "because", "since", "as", or only juxtaposition to express the justification, but the reasons are not formal in the sense of tradeoffs (Tra), decision theory vocabulary (Dt), or formal justifications (For).

   Giving a <u>desired goal</u> that an action will accomplish is an example of a reason. E.g., "To get to (a goal), I'll do this". In contrast, just to say "I <u>want</u> to do this" does not qualify as a reason; it is a "Cs".

   See discussion under Dt for a distinction between Dt and Rea.

7. Tra - The engineer discusses tradeoffs (compensations, balances) of one aspect (dimension, payoff, advantage) of an action versus another, discusses the relative advantages of competing options, uses a multi-attribute framework to consider the actions or outcomes. Prototype: "The extent to which this option is better than that option on feature A overwhelms its relative disadvantage on feature B."

8. Dt - The engineer gives reasons or justifications for his decision, using the vocabulary of probability or outcomes or decision structures (evaluated outcomes contingent on actions and events). Prototypes: "This would have a good chance of working", "If I do this, and that happens, the outcome would be good".

Some uses of the "if ... then ..." construction will fit this pattern, e.g., "If I do it this way, then the outcome will be good".

There are other uses of "if ... then ...", however, that do not fit the pattern. An example would be "if the lane width is 13 feet, that would be given a +1". This is just an application of the rule the engineer has created, it would only be called "Dt" if the engineer were evaluating the outcome. If one simply reviews the capabilities of a cognitive tool, saying "If I do this with it, it accomplishes that end", this would not be an example of Dt, for although it is a recall of knowledge about tools in "contingency" terms, it does not address the value of the outcome; rather, it only names the outcome. Such a statement, viewing a use of the tool as a possible action, would probably be classified as "Rea" or "Opt".

9. For - The engineer gives a formal justification for his decision, e.g., does a decision analysis (computing expected value, or choosing the optimum), or refers to a decision analysis or a scientific study that establishes his action to be right. Refers to a standard of measurement of how good the options are. [This category of behavior did not happen very often with these engineers making formulas.]

For Level A, code each segment according to the kind of decision-relevant activity that is most apparent in it. Thus, if you have a full sentence like "(48) If it's 13 feet it's good (49) so make that a +1", you should recognize that there is one action here along with its justification, and code 49 as Act and 48 as Rea. This orientation contrasts with that at Level B, where the coding will acknowledge whether the activity was justified at all, even if outside the segment.

If the engineer is stating his inability to take an action, inaction too is an action which could be conscious, supported by reasons, etc.

Priorities. If one segment can reasonably be coded by more than one of the Level A categories in the above list, use the later category.

Level B: The Use of Justifications

A basic element of our image of analytical reasoning (besides the notions of breaking a task into subtasks, as in Level F, and being aware of what he is doing ("Cs" at Level A)) is that the engineer can justify these steps. Level B codes whether or not the engineer is offering justifications for what he is doing. In contrast to Level A, where each segment was coded individually, here at Level B we will attempt to code at the level of whole thoughts, or sentences, or activities. If an activity was justified, then all its segments get the Jus code. If not justified, then no code at all will be used. ("An activity" is not identical with "a sentence". A long sentence might have several activities, some of them justified and some not.)

In the transcripts there are justifications of (a) actions and decisions, (b) beliefs, conclusions, uses of knowledge, (c) steps in the building of a formula, and (d) goals, shifts of attention. The present level is designed to capture all of them.

Use this level whenever the engineer is saying why an action is justified, is good or bad; why a piece of knowledge is reliable or unreliable; why he is taking a particular tack in the building of his formula; why he is paying attention to something or ignoring something. If there was an action taken, but no justification was given, do not code the segment at this level. Thus, if a segment was coded at Level A with the "Rea" category or a later one, then it (and the whole idea of which it is a segment) should be coded as a "Jus" at Level B.

1. Jus - indicates the presence of any form of justification, including just giving reasons, or using plausible or probabilistic inference, or using inductive or deductive logic. Apply the code to all segments in a justified action or clause. However, if the justification is just for one part of a coordinated action, use it only for that part. E.g., in "(5) I'll use dimension 1, (6) and dimension 2 because it is important, (7) and dimension 3", you would only code segment 6 as Jus at Level B.

Codes Pertaining to Knowledge

Knowledge and memory are essential to any cognitive activity. It seems feasible to measure three aspects of their use: (c) whether the use of knowledge involves recall of semantic or episodic memory or the use of knowledge in making a judgment; (d) whether the knowledge was produced through the heavily analytic processes of science and math, or through the person's experience (presumed to be more intuitive), or through some sort of applied science (presumed to be in between the others); and (e) whether the relations between concepts are founded on causal knowledge.

Level C: Episodic Memory versus Semantic Memory versus Knowledgeable Judgment

Use this category whenever it is apparent that the subject was drawing on knowledge or long term memory. Simply restating the givens of the problem, or recalling something one had decided earlier, should not be coded at this level.

The main distinctions among the following categories are that Ep and Se involve the statement of facts that the engineer knows, while Jd involves the use of such knowledge, its application to the present problem. "Jd" statements are made in the terms of the problem, not in some general terms; but these statements could not have been made without relying on knowledge that the expert had before.

1. Ep - Episodic Memory or knowledge of specific objects, particular places or events (as opposed to general facts). The "episode" aspect of this category represents memory of episodes or events. The events are encoded and reported in narrative form: a sequence of things that happened. For example, something that the engineer saw once that illustrates a point he is making, or supports a decision he is making. Although the engineer may draw on only one "scene from an episodic memory", you should have the sense that he could launch into a story about what else happened that day if he weren't keeping oriented to the current task.

If the engineer refers to how he solved a previous problem, e.g., the formula he worked on last week, then that is an Ep memory.

The "knowledge of particular places" aspect of this category represents memory of particular highways, intersections, curves, etc., that the engineer refers to.

2. Se - Semantic Memory. This category represents knowledge of the world, abstracted from particular events. Statement of facts, of knowledge of relationships, of general knowledge, of scientific knowledge, of memorized formulas, would be coded with this category. Include the statement of facts of arithmetic, such as "6 - 3 is 3".

3. Jd - Judgment. This category represents the use of one's long-term knowledge in making a judgment of any sort. For example, judgment of whether a dimension is important in determining capacity, or judgment of how much of the range on a dimension can be considered "equivalent" and included in the same class.

The Jd process involves both the use of knowledge and the use of categories or scales from the present task. That is, the engineer would not be thinking this thought in these terms if he was not trying to do this particular task, but he could not say what he was saying if he did not have some knowledge besides the information we gave in defining the task for him. Jd contrasts with Ep or Se, which are simply the statements of specific or general facts (respectively) in their own terms.

For example, when the engineer is thinking about different approaches he could take to this problem, and says something like "I could make that a positive number somehow, make it not 'intersections per mile' but 'miles per intersection'," he is making use of his knowledge of how to use mathematics to express relations. But he is not simply recalling facts, rather he is using this knowledge to make statements in the terms of the problem.

Use of the judgment category (Jd), instead of sweeping these instances of judgment in with semantic memory (Se), allows us to maintain the clear distinction between the recall of a fact that is related to one's construction of a formula, and the making of a judgment in order to help construct the formula. In the Se category, the recall is direct, and a recalled fact (such as the statement of a relationship) is stated. In the Jd category, the use of knowledge is indirect, the knowledge is not stated, but is simply necessary in order for the statement to have been made truthfully.

It is plausible and appropriate to say that such judgments use knowledge. The objects being judged are things the person knows about, not things that the researchers invented. To judge one, or to compare several (e.g., one dimension compared to another in terms of importance), requires the use of the engineer's knowledge about the safety of roads and about these dimensions of roads. On the aesthetics task, they are using their common sense knowledge.

Coding the segments in which a judgment of this sort is made as knowledge allows us then to determine the source of the knowledge (Level D). Most of these instances will be experience, probably. If we sense that they are completely guessing, we should not code the segment as Jd, "the use of knowledge in making a judgment".

The Jd category will be applied to any segment where the person is making a judgment about some object or concept, or about the relation between two concepts, where the judgment could not have been made unless the engineer had knowledge of the concepts. This includes knowledge of the concepts in terms of which the dimension is defined, as in "roadside culture". An engineer need not have had prior knowledge about something called "roadside culture" in order to make use of his knowledge in making a judgment about the relative strength of its aesthetics-impact in comparison to that of something called "landscaping". "Jd" would also be applied to situations where the engineer is recognizing similarities or overlap in safety-impact between two concepts, as some did with obstructions and lanewidth, for example, or shoulder width and lane width. It would also apply where the engineer is defining "equivalence classes" on a dimension, since he has to know what the numbers mean with respect to their impact on the task in order to do so.

If the engineer is looking over the formula he has so far, preparing to rethink a part of it, this need not be counted as a form of use of memory or knowledge at Level C (nor as a judgment of a quality or relation at Level G).

Level D: The Source of the Knowledge Used

The informal model here is that our culture (or this individual) has "produced" knowledge which the engineer has learned and is able to "utilize". Though the utilization may be rote, still the cognitive activity can be considered somewhat analytical because of the analysis that went into the production of the knowledge.

The knowledge that is used was produced using a particular level of cognition, we assume. Further, we assume it is used with a particular level of cognition. Finally, we assume that the cognitive mode of knowledge use is pretty close to the cognitive mode of knowledge production. Thus, knowledge from scientific fields would be produced using the best, most analytical of cognitive safeguards, and when it is used it would be used analytically. At the other extreme, knowledge based on one's own experience would be relatively intuitive, and the process of using it would also be relatively intuitive*.

1. Pu - Pure Analysis. Mathematics and mathematical tools. Logic. (most analytical)

2. Ba - Basic Science. E.g., physical principles.

----

* Note that the production and utilization of knowledge are conceptually distinct. In memo Prplan30 we proposed coding them independently. But it seems that they are highly related in practice, so we collapse the two concepts into one here.

3. Ap - Applied science. Engineering science. In this study, this would be scientific knowledge specific to highways.

4. Ex - Knowledge based on experience. In this study, this would mean experience with highways, not experience with math. (most intuitive)

Level E: Knowledge about Co-occurrences: Causality and Correlation

Use this category whenever the engineer is stating (or explicitly assuming) his knowledge of the co-occurrence relation between two concepts or variables.

When talking about the relations of co-occurrence between two concepts, variables, or events, one might or might not use causal reasoning. If the reasoning is causal, the connection may be predictive (A causes B), diagnostic (B is a sign of A), or some more complex or ambiguous pattern (e.g., A and B are both caused by C; A causes B and B causes A). [The knowledge that "A causes B" can be used predictively or diagnostically; the coder must determine which way the emphasis goes.] If the reasoning is not causal, the engineer will simply state that the two concepts are correlated (or, independent, as long as the question of a relation is being discussed).

Note that applications of parts of the formula, in numerical terms, are not directly "uses of knowledge". E.g., "If the traffic mix was 35%, it would give me a 3". This is just an application of his rule, it is not substantive knowledge, so it should not be coded at Level E.

1. Pre - Prediction. The relation between the two concepts is causal and predictive. That is, the engineer is starting with the cause and inferring the effect.

2. Dia - Diagnosis. The relation between the two concepts is causal and diagnostic. That is, the engineer is starting with the effect and inferring the cause.

3. Amb - Ambiguity. The relation between the two concepts is causal, but the direction of causality is ambiguous (e.g., A and B both being caused by C; A causing B and B causing A) or is not specified.

4. Cor - Correlation. The relation between the two concepts is not causal, is just correlational (as far as the engineer reveals in his protocol, including in the context, in segments other than the one being coded).

Codes That Address the Engineers' Thinking

This group of categorization schemes deals with various aspects of the engineers' thinking processes.

Level G: Discussion of Qualities and Relations

Qualities of items or of dimensions, and relations between pairs of items or dimensions, are basic building blocks when making a multi-attribute judgment or when producing a formula for representing one's judgments.

If the engineer is simply trying out a formula with the "highest" or "lowest" possible values on each dimension, no judgments of qualities or relations are going on, so it does not need to be coded at this level or Level H unless there is clearly a new judgment being made.

1. Qua - Statements of qualities. To say that an object is good on an attribute, or that an attribute is important, are examples of judgments of qualities. (Although these are partially analytic, because they deal with things broken down into attributes, they are also part of the structure of this task; the engineers won't be getting much more intuitive than this, so this will be our most intuitive code at this Level.) This can be nonquantitative (for example, to state that a value of a dimension is substandard) or quantitative in various degrees (e.g., to say a dimension is very important, or that it gets a weight of 9).

2. Com - Statements of comparisons. Making comparisons between the attributes of two objects, or between two attributes of an object or of objects in general. For example, to state that one dimension is more important or more variable than another; or to state that the two dimensions are identical in meaning, or will be expected to covary; or [meta-level] to state that one option for representing a formula part is better than another option.

3. Rel - Statements of general relationships. Relating two dimensions to each other in abstract, general terms, as in "the more curves, the more no passing zone" or "the more curves, the more accidents". These may be causal or acausal relations. [Note: To state a relation by saying that "the more of this one, the more of that one" is pretty intuitive, while to state it in mathematical, functional terms is more analytical. This distinction is to be captured by Level H, below. Whether causality is attributed to the relation will be addressed in Level E, above.]

   Another form of relationship is to state that one variable depends on a particular combination of two others, e.g., "safety is a function of percent no passing times traffic mix". And, finally, the statement of the whole formula is a statement of a relationship.

In deciding what function a number is performing, recognize that if the subject is prioritizing or ordering dimensions, then comparisons are being made. When the subject is assigning weights then <u>quality</u> is being assessed (within the constraints of the ordering). If the subject has not previously established an ordering when he assigns weights, then he may be simultaneously making comparisons and judging qualities; still the assignment of weights should be coded as "Qua" rather than "Com". If the subject is assigning numbers such that a unit change in one variable causes a specified change in the other variable, he is addressing relations. (There may also be the addition of constants as part of this process.) Sometimes it will be hard to distinguish these activities. You may have to go back to the start of the sequence to discover what he was trying to do.

Level H: Quantifying Qualities, Comparisons, or Relations: The Use of Numbers

This level addresses the way quantities are dealt with in a statement. It should be coded whenever Level G is used, that is, whenever a quality is judged, a comparison made, or a general relationship expressed. It should not be used when the subject is fiddling with the numerical details of the formula, unless it is clear that the subject is thinking about a quality, comparison, or relationship. Where numbers are used for other purposes (e.g., as labels), the segment should not be coded at this Level. If a statement of a quality (etc.) is broken into two segments, and only one uses quantifying terms, just code that one at this level. The other is irrelevant (ir).

1. Vag - Expression of a quality or a relation without the use of numbers. Just using vague quantitative terms, such as "a lot", "more", "very", etc. Use of "good", "important", or "substandard", etc., would also be considered "vague quantitative terms", since there is an implied ordering of such terms (from good to bad, etc.). (These are the kinds of terms that are addressed by "fuzzy" quantification.) E.g., "this is more important than that" is an example of a vague quantitative expression of a comparison. "As this gets bigger, that gets more dangerous" is an example of a vague quantitative expression of a relation - even to express the direction of relation is counted as "Vag" rather than being left blank at Level H if Level G is "Rel". To state "There is a relation between X and Y" would <u>not</u> be counted as "vague quantification", however.

2. Rat - Use of numbers to rate a quality, or the quality of several items, or to express comparisons or relations between particular things. E.g., "this has an importance weight of 5" (for a quality "Qua"), "the difference between this and that is 3" or "this is twice as important as that" or rank ordering, "this is number 1", (for a comparison), and "divide this factor by 1000 to express its impact on that factor" (for a relation).

3. Var - Use of variables to express the quantity of a judgment of a quality, or of a comparison, or of a relation. This is not to be used when the <u>dimension</u> (or a set of dimensions) is treated as a variable, only when the quantitative expression of a judgment <u>about</u> the importance (or some other feature) of the factor is treated as a variable rather than being stated as a vague quantitative expression or as a concrete quantitative expression (i.e., a number!). E.g., discussion of the comparison between two dimensions in terms of a variable: "this one is some constant times that one; I'll decide on the constant later", or "x times percent no passing plus y times traffic mix is my 'hills' factor".

   This category is not to be used when the engineer says he will now decide on his "weight factors". "Weight factors" is a term representing a class of particular factors, but this is not the desired sense of "variable".

Coding the Degree of Cognitive Control and Confidence

Other reflections of the degree of intuition are difficulty verbalizing and confidence (Hammond, 1980, 1981). We will count rather than code the occurrences of all the categories in Levels J and K.

Level J: Difficulty Verbalizing

The motivation for this set of categories is that the more intuitive the process, the more difficulty the engineer will have verbalizing it. We need a measure of how much difficulty the engineer has verbalizing. [Potential problem with interpreting these as indices of analysis or intuition: It could be that "the heavier the cognitive load (possibly, the more analytical the cognition), the worse the verbalization, due to overloading". And further, it could be that "the more non-verbal the cognition (e.g., numerical, which would be analytical), the worse the verbalization". Both of these causes would compete with the notion that "the more intuitive the cognition, the more difficulty verbalizing" in affecting the following measures.]

Components of this measure would be:

1. Pa - pauses. "1" if there is a pause in the segment. If there are pauses between segments, credit them to the following segment. Count "uh", "ah" as pauses. An "interruption", as when the subject changes a thought in midstream, is not necessarily a "pause", and will be coded as Ch or Re or Inc.

2. Re - rephrasings and repetitions. "1" if this segment is a repetition or rephrasing of a just earlier segment or of a word or phrase within the segment. Include single words that are repeated.

   In this category there is one complete idea expressed in the sentence, yet one part of it is said twice, either with the same words or with different words with approximately the same (or edited, improved) meaning. Thus, if a section rephrases the content of an earlier part of the same sentence, and then goes on to complete the sentence, it should be coded as a "rephrasing".

3. Ch - sentences that change structure in midstream, in other words, that do not end up finishing the idea that they started with, or are grammatically incorrect (as if the topic has changed). Be strict in coding deviations from correct grammar. However, if there is a plausible reading in which the sentence is grammatical, even though it be unusual (allowing for a missing comma, etc.), consider it to be grammatical.

   Implement this coding on adjacent pairs or sets of segments: code both segments "Ch" if there is incoherence between them. But if the sentence goes on from there, do not code every segment in the entire sentence "Ch" just because there was a change in structure or direction between the first two segments. In other words, if there is some structural continuity between the first segment and the second segment, call it a change (Ch) (marking both segments) rather than calling the first an incomplete segment (Inc).

4. Inc - incomplete sequence. "1" if this sequence does not end up being a complete sentence or a complete idea. (If it ends up being completed ungrammatically, or with a change in direction, count it under "Ch".) If the interrupted phrase is followed by a complete sequence (or by another incomplete sequence), then the phrase is coded as incomplete. If what follows does not stand on its own, that is, if it is a partial continuation of this segment, code them both "Ch". Incomplete parenthetical remarks should be coded as Inc, rather than coding the surrounding segments as Ch. Don't count something the engineer is reading as an incomplete sentence.

5. Mut - Muttering and inaudible verbalization. This "inaudible" category encodes words and phrases that the typist could not hear, even if the researcher could make them out on a second listen. The latter class are hand printed on the typescript. However, when the typist "cleaned up" the speech but the researcher printed in the repeated words, etc. (other than ideas the typist could not hear), this should not be considered "inaudible".

   If the inaudible speech was between the boundaries of segments, apply it to the following segment, unless the idea clearly completes the first segment.

Level K: Confidence

It is predicted that the more intuitive the process, the less confidence the subject will have in his method (Hammond, 1981; Hammond, Hamm, Grassia, and Pearson, 1987). Hence we need to count the occurrences of statements of doubt and of confidence. These will index the moment-to-moment variation in the amount of confidence. If the engineer is not confident in his knowledge of how to make a formula, or in his skill, or even in the quality of the given information, this should be counted as a doubt about the method per se, because the engineer's various kinds of knowledge and information are essential parts of the method.

1. Dbt - Doubt. "1" if the segment contains an expression of doubt or uncertainty about what to do, about the quality of what has been done so far, about the feasibility of the task, the quality of the given information, etc. Code passages that explain or elaborate on the reason for one's uncertainty as "Dbt" also.

   Posing the question of what to do next does not necessarily qualify as doubt. Evaluating an attempted solution or a proposed option does not necessarily qualify as doubt. Thus, if the engineer notices and corrects a slip, it would not be counted as Dbt, but if he raises the question of whether something is mistaken, and has trouble deciding (e.g., "I guess I'll change that to such and such"), then it would be counted as Dbt.

2. Cnf - Confidence. "1" if the segment contains an expression of confidence that the procedure is working, that it is going to work, that a part already completed can be relied on, that the task is feasible, etc.

Coding the Engineers' Self-defined Subtasks

Four categorization schemes were used to code the subtask that the engineer was engaged in.

Level I: Parts of Formulas

Level I is concerned with the activities that the engineer needs to do to make a formula. The categories are derived from the discussion of aggregation and related concepts in the Anderson, et al (1981) glossary.

1. W - think about whole formula at an abstract level, such that both the organizing principle and the dimensions are referred to. For example, to be gathering information about what the task will involve (without focussing on particular dimensions); to try to remember formulas one has written for other tasks; to remember the formula generally in use in the field; to assemble the whole formula from decisions/judgments that one had previously made, or from information about dimensions (e.g., weights) that one had previously assempbled; or to report one's final formula as a whole (even though each word refers to a specific dimension).

2. OP - think about organizing principle. E.g., the decision of whether to use a weighted averaging principle or something else. Note that the person might not say "a weighted averaging principle" but instead speak about "weights" which, by some definitions, are about dimensions; that would still be called OP. Another example of OP is to consider the general problem of how to translate from the scale of each dimension into a scale that represents the answer. However, working that out for a particular dimension would be called "dimensional".

3. Dim - think about a dimension or dimensions. This includes comparing dimensions with each other, if the point is to determine how each dimension is to be treated (within a given framework or organizing principle) rather than to determine what kind of general principle will be used as a framework for combining the dimensions together. For example, to select a specified kind of formula with a specified kind of parameter is organizing principle activity, while to select a number to serve as the actual parameter is dimension activity.

   Note that discussion of the answer dimension, though it could be considered to be a dimension, should usually be coded as Whole or Organizing Principle activity. "Whole" if the engineer says, e.g., "I have to make a formula that produces a measure of safety on a scale from 1 to 32." "Organizing Principle" if he asks, e.g., "How am I going to combine these dimensions so they produce a measure of safety on a scale from 1 to 32?"

4. Inc - Incidental conversation, e.g., about the taperecorder, or the use of the calculator.

5. Orph - Orphan. Some words or phrases are unfinished thoughts. They may be finished later, or abandoned. It is not necessary to categorize everything. If the idea gets finished later, or started over, then the essential chunk can be coded and the rest left uncoded (rather than trying

to rearrange the transcript). If necessary, one may use the uncoded chunk to discover what was meant when the sentence was finished.

If either of the last two categories is used, it should apply at all levels. Hence no categorization at other levels is necessary. However, due to the difficulties of communicating among coders, these incidental or orphan segments were often coded. They should be excluded from analysis.

Level II: Search Metaphor of Problem Solving

Level II is the search level, that is, the steps in the search process that we assume the engineer is doing in order to come up with a satisfactory way of handling the organizing principle or the dimension, or of producing the whole formula. The categories at this level are: information gathering, stating constraints, generating formulas or formula parts, evaluating either the constraints or the formula parts, and reporting the formula.

We use the informal model of formula-generation as problem solving. Problem solving processes may be understood with reference to an image of a problem space, a vast space of possible actions that could be taken, and a subset of them that are correct or acceptable solutions. Each action that one takes can be seen as narrowing the space of possible solutions, until only one is left - a completely specified formula. There are two kinds of space-narrowing action: stating a constraint and specifying a formula or formula part. We will use the C category for stating a constraint, and the G category for specifying a formula part. Evaluations can be made of C actions or G actions. One can be doing Pregenerate activity before C's, G'c, or E's. The I and R categories are not much involved in this active search process.

1. I - Information gathering. Reading the instructions, recalling relevant facts to help in generation or evaluation, or asking questions for clarification of the given information.

2. P - Pregenerational activity. Familiarization. Musing. Playing with the ideas to get a handle on them. To think about information after it has been "gathered" but before stating constraints or generating or proposing formulas or parts of formulas. Evaluating the quality, reliability, accuracy, or relevance of one's information.

3. C - State constraints, or focus. Stating a constraint involves narrowing the space of possible answers without stating a specific formula or formula part. Thus, examples are: to include or exclude a dimension, to state the relation between a variable and the answer, to say that the weight on one dimension should be larger than the weight on another, to state that a dimension is important.

4. G - Generate or explicitly propose a formula or a part of a formula, an organizing principle, a specific use of a dimension, a weight or parameter, a function form or a sign, using the hard, mathematical, technical vocabulary of formulas. So mention of a dimension is not coded "G" unless its use in the formula is specified by weights, parameters, function forms, or organizing principle.

A statement of a plan or procedure to use to produce specific formula parts (e.g., weights) is a "pregenerate" step rather than "generate".

If he is working on his formula and tries out a "best case", this is part of "evaluation" not "generation", although you would naturally say he was "generating a best case".

One should note that the delineations between "Generate" and "Constraint-setting" (or even "Pregenerate") are different, according to whether the engineer is working on an Organizing Principle or on a Dimension or the Whole formula. An Organizing Principle is an abstract thing, while the treatment of Dimensions, and of the formula as a Whole, is eventually quite specific. Hence for the D and W categories of Level 1, a Generation of a formula or formula part must be very specific (at Level 2); if not specific we call it a Constraint. But for the OP category of Level 1, an abstract statement of how the OP will behave is sufficient to qualify as an instance of Generation. The application of C to OP segments, then, will be rare, occurring only when constraints can be identified that are <u>more general</u> than a specific organizing principle.

5. E - Evaluate a possibility or "state"; that is, to think about how well a particular constraint will serve, or about how well a specific proposed formula (or part of a formula: weight, parameter, dimension, sign, function form, or organizing principle) will do. To evaluate it from one or another perspective, to consider whether to affirm or reject a constraint or formula part (or anything enumerated in G) that has already been proposed. Note that a constraint, formula or part-formula can be evaluated immediately before or after it is proposed, or much later.

   Evaluation includes both positive and negative evaluations as well as justifications. Thus, when first generating a formula part, the engineer may justify it or give positive evaluations for it, its advantages. It does not matter whether he gives the justifications or positive evaluations before or after stating the formula part, as long as it is part of the same idea. And he may evaluate a formula part negatively, whether saying "I now see that I made a mistake in proposing this" (i.e., that he should have known better) or "that proposal proves not to work" (i.e., no one could have known it was bad without trying it out).

   The Level II "Evaluate" concept does not include evaluating the quality (reliability, accuracy, relevance) of information. That should be in "pregenerational activity" (not in "gather information", because the engineer is trying to decide what <u>use</u> to make of the information).

   Do not make the error of confusing levels. That is, the activity may involve some sort of evaluation or judgment at Level III, but that does not mean it should be called "Evaluate" at Level II. For example evaluating the product of a plan or rule-guided activity with respect to the goals of that particular plan (e.g., "did I calculate correctly?") would be put in Ce, "evaluate plan or procedure" at Level III, though it might be "Generate" at Level II. Evaluating an object, captured by "Judgment" at Level III, may take place in "Pregenerate" at Level II.

6. R - Report a product or a conclusion. This category is for the engineer's "final report" of the whole formula, or for a final report of some part of the formula. Due to the requirement to communicate the formula (or to the natural pleasure one takes in an accomplishment), there will be a report that can not fairly be called "generate" or "evaluate". Such a statement would have an air of finality, of "this is what we have accomplished, let us now move on" rather than of "this is what I am thinking about, since you ask".

   If the subject is "explaining" something to the researcher, it probably is "retrospective" rather than "concurrent" at Level IV; and if the subject is "explaining" something to the researcher, all this material will be coded according to the 6 Level II codes. So the engineer could explain to the experimenter that he generated a formula part, and this explanation will be coded as G rather than R. If he says "and then I wrote this stuff on this sheet", code it R.

Priorities among Level II Categories

For many segments, more than one Level II category may seem appropriate. Perhaps there are two separate phrases in the segment, each with its own appropriate code. Perhaps a single phrase has two functions. Or perhaps it is ambiguous what is going on, or which category fits. In the cases where there are several different phrases in one numbered segment, you should pick the category that is more important or more distinct. In cases of ambiguity, go by the following priority order:

R > G > E > C > P > I

Level F: Stages of the Analytical Decomposition Process

Every one of these formula-production sessions was, by definition, an analytical process. This level of coding will attempt to describe the session in terms of the "analysis by decomposition" or "divide and conquer" strategy. The stages of decomposition are, in brief: to think about the task, to break the task into subtasks, to coordinate the subtasks, to perform the subtasks, and to combine the results of the subtasks.

In analytic judgment, it might be said that two things are produced: a method for producing judgments, and the judgment. In our particular task conditions, there was a strong demand to produce the method explicitly, and little attention was paid to specific judgments.

There are two main types of decomposition. First, the engineer may say that the concept (e.g., "safety") is a function of some other concepts (e.g., property damage accidents, injury accidents, and fatal accidents; or the ten dimensions we provided), and proceed to think about how to measure each of them. Second, he may say that to produce a formula for his concept, he needs to do first one procedure (e.g., prioritize the dimensions) and then another procedure (e.g., assign weights to the prioritized dimensions). We will apply the Level F categories to both of these kinds of decomposition, despite the fact that there are important differences between them. For example, decomposing the safety task by saying that to judge safety, one must judge lane width, shoulder width, etc., does not impose an order upon the subtasks. However, decomposing a task into a prioritizing and a weighting process requires that the prioritizing come first.

We will encode the activities into the following steps:

1. Na - Name, acknowledge, register the goal of the task (or subtask). E.g., define or redefine what the formula is supposed to do. E.g., state that you are about to rank order the dimensions to produce a prioritization. This "naming" process would include attempting to redefine the task so that it could be addressed with the analytical tools one has at hand.

   When the engineer is addressing a subproblem or subtask, turning his attention to it in order to solve it right now, then (if that subproblem gets further broken down) the segments get called "Na" from the point of view of the analysis of the subproblem. However, when several subtasks are named as part of the discussion of the main task, this should not be coded "Na" but rather breakdown ("Br") (if the engineer is saying that this is one of the things he must do, in order to solve his main task) or structure ("St") (if the engineer is considering how he will combine the results of the subtask together) or combination ("Co") (if the subtasks have already been accomplished and the engineer is now setting out to integrate them).

2. Br - Breakdown. Break the task into subproblems, subjudgments, or subtasks. Use this when the engineer says that in order to do a given task, he must accomplish a list of other tasks.

   Breakdown is conceived of as a one-time thing for any given task. After it is done, then the coordination of the execution of the list of tasks (the product of the breakdown or decomposition) is considered "structuring", except for the actual execution of each of the subtasks, which is coded "judgment" or else involves a recursion, another instantiation of the entire NBSJC pattern. Of course, the engineer can repeat a statement of a breakdown to reorient himself or to stimulate the next step of his thinking; and he could redo a breakdown differently. Either of these would be called "breakdown" (Br).

   This breakdown can be either type of decomposition mentioned above - saying that the concept is composed of a list of other concepts, or saying that the task's goal can be accomplished by doing a series of subtasks. In addition, there is a deviant kind of "breakdown", in which the engineer states that "In order to accomplish the main task, I must accomplish some other tasks..." - yet he names only one other task, not the list of two or more that we would say makes a proper "decomposition". This should be called "breakdown"; it is distinct from the "redefining" that we code as "naming", because this subtask is clearly a distinct process, not an alternative conception of the main task but a different task that contributes to the main task.

   Ambiguities may arise in applying this category. If, for example, the engineer starts doing a sequence of subtasks, without explicitly stating the breakdown, then we know two things: there has been a breakdown (Br), and he is taking hold of the first task in the list of tasks to which the original task was broken down. We can not put both of these codes onto a segment, so let us use the category that applies most directly to the segment. If the segment refers to the notion of a breakdown, to this task being done in order to accomplish the higher level task, to there being a list of such tasks that need to be done, then code it with a "Br". If all that can be seen is the subject doing the task, then the segment should be coded with Na or Ju, as appropriate. (Na is used if the subtask is subsequently broken down into a series of subsubtasks.)

3. St - Structuring. Establishing or using a symbolic structure (or superstructure) for relating the results of the subjudgments. Thus, after the engineer has determined that "supertask" can be decomposed into "subtask 1" and "subtask 2", if he discusses how the results of the subtasks will be combined, or how they will be measured, this would be considered "structuring". Just reviewing the facts, e.g, "I've got D1, D2, and D3" or "I've got to do procedure 1 and then procedure 2", might count as "establishing a structure"; it might, however, be evidence for the breakdown, Br. If the emphasis is on carrying out the subtasks (D1, D2, and D3, or procedure 1 and procedure 2), then it should be called "structuring". If the emphasis is on how accomplishing these subtasks will be a way of accomplishing the main task, then it should be called "breakdown". The actual judgments of single dimensions or pairs of dimensions would be subjudgments, Ju. Determining what scale the numerical subjudgments are to be made on, to allow for coordination among the results of the subjudgments, would be structuring, St.

   Establishing the context for the subjudgments, e.g., deciding to follow a plan of making the judgments one after another and recording them on a sheet of paper, is called "structuring" except for the segments where the subjudgment is made, where the actual subtask is performed. This distinction should be drawn at the most natural boundary. It should not be drawn so strictly that only the instant of judgment is called "subtask" (Ju); rather, the subtask as a whole should be called Ju, and only the connection and coordination (once the engineer has actually named the subtasks) should be called Structuring.

   Structuring (St) can also apply to the planning of actions whose function is to combine the results of previously accomplished subtasks (Co). That is, coordination of a number of Co segments would be called Structuring.

   A potential source of confusion is that some of our subtasks (the results of our second type of decomposition) actually create a structure for the dimensions (the results of our first type of decomposition). Thus it is ambiguous whether to call a segment "structuring" (with respect to the dimensions) or "naming" or "breakdown" (with respect to the subtask of, e.g., creating a rank ordering among the dimensions). In such cases, the segments should be coded with respect to the subtask. Thus, when the engineer is deciding he has to do a couple of processes like "prioritizing" and then "weighting" the dimensions, call that "breakdown" (Br). Once we are at this level, when he discusses how to relate the output of a "prioritizing" to a "weighting" process, that would be "structuring" (St). When he decides on the rank order or the weight of a dimension, that is the accomplishment of a "subtask" (Ju) with respect to the task of prioritizing rather than "structuring" (St) with respect to the higher level task of judging the dimensions and combining these judgments. [If he checks his work, in effect "rejudging", call that "judgment" too, unless the checking process is a distinct task that merits a whole NBSJC pattern.]

In this context, there is another ambiguity - we say that thinking about the relationship between the results of two subtasks before the subtasks are done is "structuring", but thinking about the relationship between the subtasks' results, or actually relating or combining those results, after the subtasks are done is "combining". But what should we do when the first subtask (e.g., prioritizing) is done but the second is not done yet and the engineer is thinking about how to combine their results? Call this "structuring" (with respect to the subtask yet to be done).

4. Ju - Judgments. Do the subtask. Make the (sub)judgment, or produce the lower-order part of the formula, and embody it into a symbolic statement of a relationship, quality, or quantity which can be stored, used later, referred to, evaluated. Every segment in the subtask is to be called 'Ju' only if no decomposition occurs. If the subtask is further decomposed, use the whole NBSJC pattern on it. (This is the "recursive" step. If a segment can be coded simultaneously as a Ju (from the perspective of the higher level occurrence of this NBSJC pattern) and an Na (from the perspective of the recursion, the lower level occurrence of this pattern), call it the Na from the lower level perspective.)

5. Co - Combination. Take previously encoded/remembered symbolic statements (embodying previously made subjudgments or parts of a formula or the product of a subtask) and combine them somehow (guided by one's structure) into a higher level judgment, formula, or product (which could be the final judgment or the complete formula).

   Conflict between "combining" and "doing a subprocedure involving taking the results of one previously accomplished subtask and combining them with the results of another previously accomplished subtask" may cause a coding dilemma. In cases where the coder has the option of applying either Co or Ju to a series of segments in which the subject is combining the results of previous subtasks, it should be called "combining".

This sequence of steps can be expanded (recursively) at Step 4, by replacing Step 4 with another complete cycle. We do not need to identify every step in each cycle. Steps can be arbitrarily short (even of 0 length) or long, and in arbitrary order. (We might find out that some orders do not occur.)

Level III: Information Processing

Level III is a categorization scheme of activity at the information processing level. (For coding transcripts, we use only the first level of subcategory of Control, Memory, and Judgment. The second level subcategories are listed here to provide examples.)

A. Control of cognition. Statements of goals and of steps (sequences of subgoals; plans and procedures) to attain such goals (e.g., "If I do x action, then I will attain y subgoal."); and the use of those plans and the evaluation of their results.

Some things that will not necessarily be coded as "control":

Control is ubiquitous at the information processing level of analysis. Every thought is partially controlled by previous thoughts, and partially controls subsequent ones. One poses queries, in order to get answers; one imagines situations, in order to see if some insight can be gained from them; one probes memory, in order to see if one knows a needed piece of knowledge; one considers objects in order to produce a relationship judgment or a numerical judgment. If one states an assumption, one automatically derives some of its implications.

We will not explicitly code these kinds of activity as "control", reserving this category instead for "the use of goals, plans, and procedures". Answering queries is a subclass of Judgment; imagining and probing memory are subclasses of Memory and Knowledge; considering objects is Judgment; and making deductions may be part of Knowledge, Plan Generation, or Judgment.

Things that will be coded as "control".

With respect to control through the use of plans, there are three basic types of action: generating plans or procedures; using them; and evaluating their execution or their results.

1. Cg - Generate a plan or procedure. This step involves stating, recalling or producing a plan or procedure.

   a. Goal statement. Formulating a goal or a subgoal. For example, to state that "I need to have a weight for each dimension" or to ask "What is the best way to go from here?"

   b. Plans. A plan is an informal, not necessarily complete, list of a series of steps leading to a goal. It may involve statements of contingent relations, of temporal order of steps, or of what to do in certain cases. For example, "In order to attain my goal of x, I need to do y" or "If the outcome of z is too big, I will do w".

      Working out the implications of a general statement or plan, i.e., how it constrains one's subsequent actions, would be included here. For example, if someone decides to partition 100 among the dimensions as weights, and notes that if the dimensions were all equal they would get 10 each, he is making inferences or deductions from the general statement of his plan, for a more specific statement of this plan. [At Level II, it would be G; at Level I, O.]

   c. Procedures. A procedure is more specific than a plan. It is like an algorithm or a computer program.

Part of generating a goal or plan is the process of prospectively evaluating whether it seems like a good idea. Evidence of this kind of thinking should be coded as Cg rather than Ce.

Another special category of Cg is remarks made in order to keep talking when no clear plan has come to mind yet. The engineer is trying to produce a plan. Each sentence may be a memory, or a judgment; but if it is obvious that he is trying to produce a plan, we should try to encode one of the sentences with Cg.*

2. Cu - Use a plan or procedure.

   a. Using a plan. This is following the plan that the subject made up (or retrieved from memory) earlier.

   b. Using a procedure. Carrying out a step by step procedure. For example, calculating something or producing a final statement of a formula by drawing the weights from one table and the function forms from a sheet of paper.

      It is possible that the engineer follows a plan although we did not see him generate it. Perhaps he just recalled it from memory. Numerical calculation is an example.

3. Ce - Evaluate the execution or outcome of a plan or procedure. This happens after it has been used. Use when the subject recognizes he has made a mistake, for example, or when he is doing a parallel procedure to check whether everything came out all right.

   This category is designed to cover both "slips" and "mistakes", in Don Norman's terms.

   a. Slips. If he discovers he has slipped in doing what he intended, then the evaluation is at the level of the plan or procedure, at the level of control.

   b. Mistakes. If he discovers that the result of the plan does not contribute to a good formula, that is, if he is thinking about it with respect to a formula in particular, rather than as a procedure in general, then it is a "mistake". It should be coded as an E at Level II. At Level III, it should be coded as Ce, unless it clearly involves:

      1. judgment of relations between entities or dimensions - use Jv

      2. numerical judgment of qualities - use Jn

---

* Having a policy allowing the application of several categories to the same sentence or unit would remove the need for this particular kind of fallible and idiosyncratic judgment, but would introduce the need for such judgment in many more places. I think the problem of deciding "which category is most appropriate in this context" is easier than deciding for each of a number of categories whether or not to include it.

An evaluation (II-E) of a proposed formula part would also be an evaluation of the execution of a plan (Ce) if the subject had to do several steps in order to construct the formula part, and then saw that it would not work, yet does not express an explicit judgment in announcing this conclusion.

B. Memory and Knowledge.

This category addresses the subject's use of his memory and knowledge in two important ways. First are the simple actions of storing and retrieving knowledge, elementary activities on which so much else depends. Second is the use and elaboration of knowledge of the world in order to discover things that are needed for building a formula but which have not been thought about in exactly these terms before.

1. Ms - Store in memory. This category is to be used only when the storage in memory is the most important function going on. We assume that the engineer stores almost everything he says, every new idea he generates, every judgment he makes, every plan he wants to follow. But storage is not the main point of each of those actions. For example, after the subject reads all the information at the beginning (which we would indeed say is storing in memory), if he reads something later from those sheets, he is probably getting it for immediate use - he is recalling it from external memory (he remembered where it was, and he got it when he needed it), rather than storing it. When the purpose of the activity is to store information, we should code it as 'memory'. Also, if nothing else is going on we should code it as 'memory'.

   There is a special use of Ms, when the engineer Generates a formula or formula part, decides to do it, and simply states this decision, remembering it for later use. This can be distinguished from Cg, generating a plan to be followed step by step right now. For these "Ms" cases involve the remembering of a contingency, a rule to be applied later when appropriate. Note that some cases which seem to deserve being coded "Ms" for this reason will actually be called "Cu" because the statement completes a plan.

   a. Use rehearsal strategy for maintaining information in STM

   b. Use rehearsal or association strategy for storing in LTM (If this or the previous is not done consciously, there is still some fallible automatic storage.)

   c. Write for storage in External Memory

   d. State things that are of some importance and will be referred to later, e.g., intermediate products in a calculation (if we code calculation in such fine detail) or parts of formulas. This includes writing things, and speaking them as he writes (or, not speaking them, if we can tell he is writing.).

Note: Sometimes at the moment the engineer is generating a formula part it is difficult to say what activity he is doing at the information processing level; it may not be a judgment nor the following of a plan. In these cases, it is appropriate to call it Ms at Level III, for surely the proposed formula part is being stored in memory.

2. Mr - Retrieve from memory.

Retrieval from memory, like storage in memory, is a low priority coding category. If the subject is doing something more directly relevant to the production of a formula (i.e., calculating or planning or judging), then code it with that other category. The idea of retrieval from memory should particularly be used when it is an activity that the subject seems to be devoting himself to, rather than it just happening automatically. [Since we are not simulating the cognitive processes, we do not need to capture every instance of recall.]

a. Use what was available in STM

b. Recall something from LTM

1) Use information that was not in external memory nor in STM nor could have been "judged" (If this is so, we can infer that LTM was used).

2) Set up a probe for LTM recall

a) Successful probe, e.g., a question and an answer.

b) Unsuccessful probe; stall

c. Read information previously written or previously registered, noted, attended.

3. Mi - Imagining. Building and manipulating images to serve as objects of one's thinking and judging. For example, if an engineer tries to think of the kinds of road that have bad capacity, and thinks of South Parker Road and describes it in detail, we would say he is retrieving from memory, Mr. (He might or might not derive insight into the meaning of the dimensions from his exercise.) But if he simply constructs an image of a "worst case" road (guided in his image-constructing by specific dimension values) or if he tries to think of the kinds of events or features that decrease a road's capacity (slow vehicles on steep narrow roads), then he is not really recalling any particular memory, rather he is constructing a new image. This new image will be drawn from, and consistent with, his knowledge.

C. Use judgment.

The judgment processes involved in translating from knowledge of highways into a formula (particularly judgment of highways or judgment of dimensions) are to fit in this category. There are two major classes of judgment subsumed in this category. A subject (1) judges a quality of an object or judges the direction of relation between dimensions; or (2) makes numerical judgments about the degree of a quality or a relation. The evaluation of formulas or parts of formulas is captured by II-E (though it may also involve this category of judgment). Other Level III categories are intended to be used for the evaluation of plans prospectively (Cg) or retrospectively (Ce).

1. Jv - Make a verbal statement of a judgment of a relationship or a quality, or set up to make the judgment. For example, the judgment of whether more of a dimension increases or decreases the criterion; or whether the relation is linear or nonlinear. If a verbal judgment is repeated, after having been said before, this should be treated as a judgment rather than an instance of memory retrieval, because it is as a judgment that it is being used*.

   If the engineer is "prioritizing a list of dimensions", he is making order judgments among them, judgments of relations, and so this should be coded as Jv, even if the engineer assigns the dimensions ranks of 1, 2, etc. For the numbers are being used as an ordinal scale.

   a. Judge an object already in STM.

   b. Set up for a judgment of a quality or a relationship. As long as the fact of the judgment gets noted, the "setting up" may also be "generating a plan" or "following a plan" or "retrieving from memory" - these categories can be used. But if there is only one phrase that accomplishes all these functions, then it should be coded as a "judgment".

      1. Describe the object to be judged. Ask the relation between a dimension and the answer.

      2. Describe anchor points

---

* This is arguable. If we were trying to simulate, or measuring how long processes take, it would be necessary to code this as a memory retrieval (if only STM) rather than as a judgment. I justify not doing so, in part, on the grounds that we can not tell whether the subject is remembering or judging anew or even is recalling the previous judgment and also judging again to check it. If we could discriminate these possibilities for sure, it would be good to separate these categories; since we can not, we should use the "judgment" category because it addresses the function the cognition serves. The possible bad effect of this is that we might slightly overestimate the amount of "judgment as opposed to analytical processes or memory" that the engineer engages in, in making our argument that "judgment" is an important category of cognition.

3. Describe calibration points.

c. Try to make a judgment, but fail to have an answer come to mind.

2. Jn - Make a numerical judgment. Examples are the invention of a numerical weight parameter, or of a cell value in a table representing the impact of the conjunction of particular levels on each of two or three dimensions. Note that if the object being judged is a number, as the "0" end of dimension, but the judgment is qualitative, as in "the 0 is the low end of the scale", it should be coded as a verbal judgment not a numerical one.

   a. Translate from a judgment or statement of a relationship into a numerical scale.

   b. Use the number scale to express the relationship or the quality in the first place.

      1) Judge an object already in **STM**.

      2) Set up for judgment

         a) Describe the object to be judged.

         b) Describe the judgment scale

         c) Describe anchor points

         d) Describe calibration points.

   c. Try to make a numerical judgment, but fail to have an answer come to mind. Talk about why it is not working, e.g., how the object would have to be more precisely specified, why the number scheme won't work (e.g., lack of anchor or calibration scheme).

Priorities among Level III Categories

Obviously there are problems with overlap among categories at Level III. To help the coder decide which category to use, the following is the priority order: If a unit can be equally plausibly described as two categories, then the one higher in this ordering should be used:

Jv, Jn > Cg, Ce, Mi > Cu > Mr > Ms

Activities in the same level of priority ordering are assumed to be clearly differentiable on other grounds.

In words, the judgment of objects or relations is considered the most important category to note, then the generation and evaluation of plans and the use of imagination; so if the engineer uses judgment in creating a plan, and there are not separate units that can be labeled with both categories, call it J. The use of plans is considered less important, so if the engineer uses judgment in following a plan, call it J; etc.

The problem of questions. Questions are asked to control what one is thinking about. We code them according to the purpose of the activity. If one asks the question to help formulata a plan of action, it is Cg. If one asks the question as part of the use of one's memory to recall information, it is Mr; if to stimulate one's imagery knowledge, it would be Mi; to set up for a judgment of relations, it is Jv.

Level IV: Kind of Verbalization

The subject's verbalization can be considered a concurrent or retrospective report of his cognitive processes (Ericsson and Simon, 1984). Level IV indicates whether the report is concurrent or retrospective, and can be left blank if concurrent (default), for retrospective reporting is rare, occurring only when the subject is explaining the process to the researcher, which occurs following researcher's queries and nudges.

1. Con - Concurrent. The directions in this task were for concurrent verbalization, and most sentences are this. The subject is thinking and saying what is heeded.

2. Ret - Retrospective. But sometimes the subject speaks retrospectively. This happens most often when he is explaining something to the researcher. For example, if the researcher probes or nudges the subject, who responds with a very high level explanation of what he is trying to do, it should be regarded as an explanation rather than a "report" or the generation or use of a plan (though it may quickly turn into one of the latter). It should be coded as retrospective when it is recognized that the subject is responding to the researcher, that what he is talking about is not taking place right now.

   This material is valuable. A prettified account of the following of a plan, it may make the subject's intentions and plans more clear than they are in the concurrent verbalizations. Also, the plan improves when the subject goes over it, so it will be more available next time he needs it, and also probably will be a better plan.

   Sometimes if the material is retrospective it may not be possible to say what kind of information processing is going on. If so, it is okay to leave Level III blank.

In cases where retrospective reporting is rare, the concurrent category can be used as a default, need not be written for every unit.

Reliability of Coding

In order to determine the reliability of the coding, selected sessions were coded twice. As the levels were divided into four sets, each coded by a different coder, it was possible to select different sessions for the reliability checks for each coder.

1. Levels I, II, III, and IV were coded by RL, and RH checked sessions 10c, 12e, and 15s. See Table 1.

Insert Table 1 about here

---

2. Levels A, B, and F were coded by CC, and RH checked sessions 1c, 5e, and 8s. See Table 2.

Insert Table 2 about here

---

3. Levels C, D, E, G, and H were coded by RH, and JG checked sessions 8s, 10c, and 12e. See Table 3.

Insert Table 3 about here

---

4. Levels J and K were coded by JG, and RH checked sessions 5s, 8s, and 10c. See Table 4.

Insert Table 4 about here

Rather than giving the reliability data for three sessions for each coder, the "median" session for each was chosen, i.e., the one for whom the reliabilities were the median for the largest number of "percent agreement" and "category correlations" categories. This allows the interdependencies among the reliability data for each session to be preserved. [The reliabilities in Hamm (1985) were the median <u>for each category</u>.]

The tables on the following pages present the following features of the coding reliability:

1. Coder %. The percent of segments that each coder gave each code to, at each Level, allows us to see whether the two coders used the category about the same number of times. Note that the Inc and Orph categories of Level I are identically applied to Levels II, III, and IV, so the redundant information is not included. For Levels JK, the number of

segments coded with a value other than 0 is given.

2. Percent Agreement. The number of times that the two coders gave the same category at a given level. For Level I, Inc and Orph are included; for Levels II to IV and A to K, they are excluded.

3. Category Correlations. These show the extent of agreement between the coders for each category. New variables were created for each coder and each category, at each level, and assigned a value of '1' if this level had the category in question, otherwise '0' (or missing if no coder choice was involved, e.g., if the segment was not coded (as for ABF, CDEGH, and JK when the subject was only reading, or for II and III if it had already been coded as 'inc' or 'orph')). Then the correlation of these variables for the two coders was determined.

4. Correlation between the subindices of the MBMCCI. Translation into the subindices of the MBMCCI is done by applying the relevant parts of the MBMCCI formula to the subindices (A, B, C, D, E, G, H, J, K). This is done for each level, and for each coder. Correlating these shows how much agreement there is with respect to the measurement of moment by moment variation in analysis and intuition.

   a. Each level

   b. Overall

References

Anderson, B. F., Deane, D. H., Hammond, K. R., McClelland, G. H., & Shanteau, J. C. (1981). Concepts in judgment and decision research: Definitions, sources, interrelations, comments. New York: Praeger.

Ericsson, K. A., & Simon, H. A. (1984). Protocol analysis: Verbal reports as data. Cambridge, MA: MIT Press.

Hamm, R. M. (1985). Moment by moment variation in the cognitive activity of experts (Technical Report No. 257). Boulder: University of Colorado, Center for Research on Judgment and Policy.

Hamm, R. M. (1988). Moment by moment variation in experts' analytic and intuitive cognitive activity. IEEE Transactions on Systems, Man, and Cybernetics, SMC-18(5).

Hammond, K. R., Hamm, R. M., Grassia, J., & Pearson, T. (1987). Direct comparison of the efficacy of intuitive and analytical cognition in expert judgment. IEEE Transactions on Systems, Man, and Cybernetics, SMC-17, 753-770.

Table 1

Reliability of Coding of Categorization Schemes, Levels I to IV

Engineer: 10c
Coders: RL and RH
Number of Coded Segments: 238

Categories		Coder 1 %	Coder 2 %	Percent Agreement	Category Correlations
Level I				88.7	
	w	11.3	7.1		.72
	op	5.0	5.9		.60
	d	77.3	82.8		.71
	inc	0.8	0.4		
	orph	5.5	3.8		
Level II				70.6	
	i	8.4	11.8		.80
	p	38.7	34.9		.78
	c	22.3	11.8		.52
	g	15.5	31.1		.52
	e	5.9	3.8		.63
	r	2.9	2.5		.92
Level III				46.2	
	cg	13.4	10.5		.53
	cu	24.8	41.6		.18
	ce	2.9	2.9		.56
	ms	8.4	16.8		.67
	mr	18.1	8.4		.13
	mi	0	0		~
	jv	16.8	8.4		.67
	jn	9.2	7.1		.02
Level IV				96.2	
	Con	93.7	95.8		
	Ret	0	0		

Table 2

<u>Reliability of Coding of Categorization Schemes, Levels A, B, and F</u>

Engineer: 1c
Coders: CC and RH
Number of Coded Segments: 359

Categories	Coder 1 %	Coder 2 %	Percent Agreement	Category Correlations	MBMCCI Each Level	Subindices Overall
Level A			54.3		.26	.42
act	55.2	34.5		.40		
cs	28.1	31.8		.48		
opt	4.7	17.8		.27		
ev	1.4	5.8		.27		
com	2.8	3.9		-.03		
rea	7.5	5.3		.45		
tra	0	0		~		
dt	0	0.6		~		
for	0	0		~		
ir	0.3	0.3				
Level B			86.9		.48	
jus	14.2	15.0		.47		
ir	85.8	85.0				
Level F			67.4			
na	18.4	18.9		.76		
br	7.5	6.1		.68		
st	5.6	18.1		.14		
ju	68.5	45.4		.62		
co	0	11.1		~		
ir	0	0.3				

Table 3

Reliability of Coding of Categorization Schemes,
Levels C, D, E, G, and H

Engineer: 12e   Coders: RH and JG
Number of Coded Segments: 92/80

Categories	Coder 1 %	Coder 2 %	Percent Agreement	Category Correlations	MBMCCI Each Level	Subindices Overall
Level C			65.5		~	.52
jd	29.3	42.5		.10		
ep	0	2.5		~		
se	0	15.0		~		
ir	70.7	40.0				
Level D			71.8		~	
pu	0	26.2		~		
ba	0	0		~		
ap	0	0		~		
ex	29.3	33.8		.65		
ir	70.7	40.0				
Level E			97.3		-.02	
pre	0	1.3		~		
dia	0	0		~		
amb	0	0		~		
cor	2.2	0		~		
ir	97.2	98.8				
Level G			79.1		.57	
qua	30.4	31.3		.51		
com	2.2	1.3		.70		
rel	7.6	21.3		.48		
ir	59.8	46.3				
Level H			83.6		.93	
vag	18.5	31.3		.63		
rat	16.3	21.3		.76		
var	0	1.3		~		
ir	65.2	46.3				

Table 4

Reliability of Coding of Categorization Schemes, Levels J and K

Engineer 10c.
Coders: JG and RH.
Number of Coded Segments: 212

Categories	Coder 1 %	Coder 2 %	Percent Agreement	Category Correlations	MBMCCI Each Level	Subindices Overall
Level J					.81	.81
pa	34.9	35.6	97.2	.97		
re	10.8	4.6	90.1	.33		
ch	3.3	9.3	89.2	.11		
inc	2.8	11.1	90.6	.36		
mut	22.2	23.6	92.5	.82		
Level K					.50	
dbt	0.5	1.9	98.6	.50		
cnf	0	0	100.0	~		

APPENDIX A

Categorization Schemes Pertaining to the Degree of Intuition or Analysis

Level A. Decisions and decision making (dm).

- Act - action without justification
- Cs - conscious of taking action
- Opt - describes options
- Ev - evaluate or negate an action or option
- Com - dm involving comparison without discussion
- Rea - dm involving simple reasons and justifications
- Tra - dm involving tradeoffs among dimensions
- Dt - dm involving reasons in terms of probability, value, contingencies
- For - dm involving formal justification - science or decision theory (expected value)

Level B. Justifications.

- Jus - use of any form of justification

Level C. Kind of Memory.

- Jd - use of memory in making a judgment
- Ep - episodic memory
- Se - semantic memory

Level D. Source of Knowledge Used

- Ex - Experience
- Ap - Applied science, engineering
- Ba - Basic science
- Pu - Pure analysis: mathematics and logic

Level E. Knowledge about Co-occurrences: Causality and Correlation

- Cor - correlation, non-causal
- Amb - ambiguous causal relation
- Pre - prediction, causal
- Dia - diagnosis, causal

Level G. Qualities and relations.

- Qua - stating judgments of qualities
- Com - making comparisons
- Rel - stating relationships

Level H. Quantities, numbers.

Vag - using vague quantitative terms to express qualities or relations
Rat - numbers as ratings
Var - variables and formulas

Level J. Difficulty verbalizing.

Pa - pauses and hesitations
Re - rephrasings and repetitions
Ch - changing sentence structure in midstream
Inc - incomplete sentences
Mut - muttering and inaudible verbalization

Level K. Confidence

Dbt - expressions of doubt
Cnf - expressions of confidence

APPENDIX B

Categorization Schemes Pertaining to the Subject's Self-defined Task

Level I. Judgment Analysis.

W	Whole formula	
OP	Organizing Principle	
D	Dimension	
Inc	Incidental remark	[no other codes needed]
Orph	Orphan	[no other codes needed]

Level II. Search.

I	Information gathering
P	Pregenerational activity, familiarization
C	Constraint setting, focussing
G	Generate a formula or formula part
E	Evaluate or justify a formula or formula part
R	Report a formula or formula part

Priority order among Level II categories, in case of ambiguity:

R > G > E > C > P > I

Level F. Stages of Analytic Decomposition Process.

Na - Name, register, redefine the goal.
Br - Break judgment task into smaller tasks.
St - Establish structure for relating subtask results.
Ju - Make the subtask judgment and state it, remember it.
Co - Combine subjudgments.

APPENDIX C

Categorization Schemes Pertaining to Information Processing

Level III. Information Processing.

Control

Cg	Generate plans, goals, or procedures
Cu	Use plans
Ce	Evaluate plans or their results

Memory

Ms	Store in memory
Mr	Retrieve from memory
Mi	Imagine

Judgment

Jv	Verbal judgment, judgment of relations or qualities
Jn	Numerical judgment

Priority order among Level III categories, in case of ambiguity:

Jv, Jn > Cg, Ce, Mi > Cu > Mr > Ms

APPENDIX D

Categorization Scheme Pertaining to Subject's Perspective

Level IV. Type of Verbalization

Con	Concurrent verbalization
Ret	Retrospective report