ED 128 420 TM 005 605

AUTHOR Christal, Raymond E.; Weissmuller, Johnny J.
TITLE New CODAP Programs for Analyzing Task-Factor Information.
PUB DATE [Sep 75]
NOTE 18p.; Paper presented at the Annual Conference of the Military Testing Association (17th, Fort Benjamin Harrison, Indiana, September 15-19, 1975); Also included in TM 005 585

EDRS PRICE MF-$0.83 HC-$1.67 Plus Postage.
DESCRIPTORS *Computer Programs; Job Analysis; *Occupational Information; Performance Factors; Rating Scales; *Task Analysis; Test Reliability
IDENTIFIERS CODAP; *Comprehensive Occupational Data Analysis Programs

ABSTRACT

Several new programs have been added to those of the Comprehensive Occupational Data Analysis Programs (CODAP), all oriented toward analyzing and manipulating information describing work tasks, rather than jobs or persons. REXALL analyzes the inter-rater agreement among judges concerning task-factor ratings. TSKFAC adds factor weight vectors to the JOB DESCRIPTION file. FACSTD generates a file of rater background information and raw ratings. FACSPC computes, displays, and saves mean factor ratings for identified groups of raters, COMGEM generates and saves factors and factor composites through user-specified operations. TSKCOR extracts vectors from the JOB DESCRIPTION file and computes intercorrelations and user-specified regression equations. PREFAC applies equation weights and produces a predictor score vector. TSKCAT generates a user-specified vector of 1's and 0's to identify tasks of interest. FACSUM selects, sorts, and prints task-factor information, and computes new vectors based on differences between vectors, maximums or minimums of vector sets, cumulative percentages, and categories of tasks developed by six arithmetic operators. These programs fill an important gap by equipping the analyst with tools for addressing new problem areas. (Author/BW)

NEW CODAP PROGRAMS FOR ANALYZING TASK-FACTOR INFORMATION

By

Raymond E. Christal
Occupational and Manpower Research Division
Johnny J. Weissmuller
Computational Sciences Division
Air Force Human Resources Laboratory (AFSC)
Lackland AFB, Texas

I am making the assumption that most of your here are familiar with the CODAP analysis system, which is a comprehensive set of computer programs for analyzing and reporting occupational information collected with job inventories. The CODAP system presently contains approximately thirty-five main programs which are oriented toward producing task-level descriptions of the work performed by individuals and groups of individuals. The system also contains a number of programs for identifying and describing the types of jobs existing in an occupational category, and for describing the characteristics of individuals falling into special or job-type groups. Today, I would like to describe a new series of programs recently added to the CODAP system which are oriented toward analyzing and manipulating information describing work tasks, rather than jobs or persons. These programs fill an important gap by equipping the analyst with tools for addressing new problem areas.

Before describing the new programs, I should recognize that CODAP already contains some programs which could be considered as being task-oriented. For example, ASFACT is designed to summarize task-factor rating information provided by job incumbents, while GRPSUM can be applied to display the percent of members in various groups performing any given task. AVALUE and AVGPCT can be used to compute such indexes as the average grade level or time-in-service of individuals performing each work task in an inventory. Two other task-oriented programs in the UNIVAC 1108 version of CODAP are REXALL and TSKNDX. These will be described briefly, since they interact with the new programs now being released.

REXALL is a very powerful and flexible program for analyzing the inter-rater agreement among judges concerning task-factor ratings. This program also generates a task-factor weight deck for input into other CODAP programs. An example output from a REXALL run is given in Table 1. This table displays certain information concerning task-factor ratings provided by each of 38 raters participating in a recent task

Table 1. REXALL Rater Correlations with Sample Means--Raw Responses

Rater ID	r_{xy}	N Ratings	Mean	S.D.	T-Value
1	.735	413	3.57	1.06	19.12
2	.655	414	4.37	0.92	15.30
3	.777	414	3.77	0.96	21.79
4	.658	414	3.56	1.08	15.43
5	.636	414	4.16	1.58	14.55
6	.697	414	2.96	1.00	17.16
7	-.474	413	4.59	1.46	-9.48
8	.791	414	4.54	0.88	22.84
9	.730	414	4.34	1.06	18.87
10	.690	414	5.09	.72	16.84
11	.031	414	4.04	1.01	0.55
12	.789	414	3.93	1.14	22.74
13	.614	414	4.07	1.16	13.73
14	.819	414	4.66	1.51	25.21
15	.639	414	3.82	1.29	14.68
●	●	●	●	●	●
●	●	●	●	●	●
●	●	●	●	●	●
38	.805	414	4.49	0.96	23.95

difficulty analysis. The program first computes the mean rating for each task across all raters; then it correlates the ratings provided by each particular rater with the vector of mean ratings obtained from the total sample. These correlations range from .61 to .81. The values provided by rater #7 correlate -.47 with the ratings provided by the complete sample. He used the rating scale upside down. This is not unusual. We have analyzed task-factor information from hundreds of raters on a variety of task factors, and we find that about 5% of the participants reverse the rating scale. Rater #11 is "non-cooperative." It is inconceivable that there would be no agreement between a supervisor and his peers concerning the relative difficulty of tasks in his occupation. This is an example of an individual who simply "filled in the squares" with random numbers. Both of these individuals contribute "error" variance to the mean factor vector, and, at AFHRL, we would elect to discard them from the data set. Another source of error is reflected in column 4, which reports the mean of ratings provided by each rater. In this particular study, the raters were asked to rate the relative difficulties of tasks in their occupation using a 7-point scale. The values in column 4 reveal that some raters tended to use the upper part of the scale, while others tended to use the lower part. For example, rater #6 had a mean of 2.96, while rater #10 had a mean of 5.09. Since all ratings are relative, there is no reason why the mean rating across all tasks should vary significantly from rater to rater. Such differences in means introduce within-task variance, which is treated as error variance in computation of the inter-rater agreement coefficient. REXALL provides a method of eliminating such error variance by standardizing ratings.

Table 2 presents summary data associated with the first REXALL run in the example. Notice that the reliability coefficient for the means based upon all 38 raters is .939, which is not bad, considering that: (a) two raters were included who reversed the rating scale; (b) one rater was non-cooperative; and (c) the ratings were not standardized. The sample value is only .288, which can be loosely defined as indicating the reliability for a single rater. It is this value which is entered into the Spearman-Brown prophecy formula to estimate the reliability of mean ratings based upon a sample of any given size.

You will notice that REXALL provides a printout of all data entering into the computation of the reliability coefficient. It also indicates, among other things: the average number of ratings per task; number of raters; number of tasks; total number of task ratings; percent of possible ratings; mean and standard deviation of all ratings; mean and standard

Table 2. Inter-Rater Reliability Based on Raw Ratings

RKK	=	.93892	Reliability for These K Raters
R11	=	.28806	Reliability for a Single Rater
F11	=	16.3724	F-Ratio (BTMS/WMS)
DFB	=	313.	Degrees of Freedom (Between, N1)
DFW	=	11616.	Degrees of Freedom (Within, N2)
DFT	=	11929.	Degrees of Freedom (Total)
C	=	199970.	Correction Factor for Sum of Sqs
BSS	=	6560.	Between Sum of Squares
WSS	=	14871.	Within Sum of Squares
TSS	=	21431.	Total Sum of Squares
BTMS	=	20.96	Between Tasks Mean Squares
WMS	=	1.28	Within Tasks Mean Squares
K	=	37.99	Average Numbers of Raters per Task
NRATER	=	38.	Total Number of Raters
NTASK	=	314.	Number of Tasks to be Rated
NZTASK	-	314.	Number of Tasks Actually Rated
NRESP	=	11930.	Total Number of Ratings
PCTRSP	=	99.98	Percent of Total Possible Ratings
TOTMN	=	4.0941.	Mean of All Task Ratings
TOTSD	=	1.3403	Standard Deviation of All Ratings
AVEMN	=	4.0942	Mean of All Task Means
AVESD	=	1.1061	Mean of Task Standard Deviations
SDMN	=	7416.	Standard Deviation of Task Means
SDSD	=	.1518	Standard Deviation of Standard Deviations

deviation of task means; and mean and standard deviation of task standard deviations.

Normally, REXALL is executed at least twice to process data for a particular factor in a study. First, it is run on raw data in order to identify non-cooperative and upside down raters for discard. In the second run, the values from the retained raters are standardized, unless an absolute scale is being used. Table 3 presents the summary information for the second run of the example data. Notice that three raters were discarded, leaving 35 raters in the run. The raw scores for each rater and for the means across all raters were standardized to a mean of 5.00 and a standard deviation of 1.00. These actions raised the reliability of the mean ratings from .939 to .973, even though the number of raters was reduced. At the same time, the sample value was raised from .288 to .503.

One of the outputs from the REXALL is a weight deck containing the mean rated value for each task in the job inventory on the factor in question. This weight deck is in proper format for direct input into other CODAP programs. For example, mean task difficulty weights might be entered into the VARGEN program to compute the average difficulty of tasks performed by each worker in a sample. Normally, however, the weight deck is first entered into the TSKNDX program in order to display the mean ratings of each task, along with other relevant information. Table 4 presents an abbreviated output from a TSKNDX run for tasks in the Electrical Power Production career ladder. In this specialty, draining condensate from air receiver tanks is considered to be the simplest task, while repairing electronic governors is considered to be the most complex. The TSKNDX program is flexible. At the requestor's option, it will publish data in task order or from low to high or high to low on mean factor ratings. It also presents other data of interest, such as the number of workers in a sample of interest performing each task, as well as the percent of time being spent on each task.

ASFACT, AVALUE, AVGPCT, REXALL, and TSKNDX are task-oriented programs which have been available for a number of years in the IBM 7040 and UNIVAC 1108 versions of CODAP. All five of these programs are currently coded in FORTRAN V, and could easily be converted for execution on other computers. While these programs are all very useful and efficient, recent studies in job evaluation and curricula development have required analyses and report formats of task-factor data which could not be serviced by existing CODAP programs. For this reason, a new series of task-oriented programs has been written to fill the gap. These are enumerated in Table 5.

Table 3. Inter-Rater Reliability Based on Adjusted Ratings

RKK	=	.97251	Reliability for These K Raters
R11	=	.50274	Reliability for a Single Rater
F11	=	36.3827	F-Ratio (BTMS/WMS)
DFB	=	313.	Degrees of Freedom (Between, N1)
DFW	=	10641.	Degrees of Freedom (Within, N2)
DFT	=	10954.	Degrees of Freedom (Total)
C	=	274728.	Correction Factor for Sum of Sqs
BSS	=	5681.	Between Sum of Squares
WSS	=	5308.	Within Sum of Squares
TSS	=	10989.	Total Sum of Squares
BTMS	=	18.15	Between Tasks Mean Squares
WMS	=	.50	Within Tasks Mean Squares
K	=	35.00	Average Number of Raters per Task
NRATER	=	35.	Total Number of Raters
NTASK	=	314.	Number of Tasks to be Rated
NZTASK	=	314.	Number of Tasks Actually Rated
NRESP	=	10989.	Total Number of Ratings
PCTRSP	=	99.99	Percent of Total Possible Ratings
TOTMN	=	5.0000	Mean of All Task Ratings
TOTSD	=	1.0000	Standard Deviation of All Ratings
AVEMN	=	5.0000	Mean of All Task Means
AVESD	=	.6847	Mean of Task Standard Deviations
SDMN	=	1.0000	Standard Deviation of Task Means
SDSD	=	1.0518	Standard Deviation of Standard Deviations

Table 4. CODAP: Task Difficulty Index (TSKDIF) for Electrical Power Production (N=530)

D	TSK	Duty/Task Titles	Task Sequence Number	Task Difficulty Index	Number of Members Performing	Percent of Members Performing	Average Percent Time Spent by Members Performing
I	10	Drain Condensate from Air Receiver Tanks	1	2.15	269	50.75	1.49
.	.	. . .	.	.	.	.	.
.	.	. . .	.	.	.	.	.
H	3	Adjust Engine Controls After Starting	119	3.82	352	66.42	1.77
L	36	Take Firing Pressure Readings	120	3.82	170	32.08	1.03
M	21	Replace Water Circulating Pumps and Valves	121	3.82	69	13.02	0.67
.	.	. . .	.	.	.	.	.
.	.	. . .	.	.	.	.	.
H	5	Compute Power Produced by Generator	198	4.22	208	39.25	1.37
M	2	Adjust High Water Temp. Alarms & Shut Down Equip.	199	4.22	147	27.74	1.08
N	6	Correct Causes of Brush Sparking or Arcing	200	4.22	155	29.25	1.02
.	.	. . .	.	.	.	.	.
.	.	. . .	.	.	.	.	.
J	8	Determine Cause of High or Low Oil Pressure	300	4.60	214	40.38	0.97
L	29	Replace Piston and Connecting Rod Assembly	301	4.60	108	20.38	0.76
L	32	Replace Rod and Main Bearings	302	4.60	95	17.92	0.72
.	.	. . .	.	.	.	.	.
.	.	. . .	.	.	.	.	.
P	4	Analyze Static Battery Charging Diagrams	407	5.38	41	7.74	1.06
M	4	Determine Cause of Hydraulic Governor Malfunctions	408	5.42	110	20.75	0.85
O	1	Adjust Automatic Start Systems	409	5.42	103	19.43	0.92
.	.	. . .	.	.	.	.	.
.	.	. . .	.	.	.	.	.
O	15	Inspect, Replace, or Repair Governor Computer	434	5.85	31	5.85	0.51
O	20	Repair or Replace Transistorized Switching Devices	435	5.85	38	7.17	0.63
P	6	Analyze Transistorized Switching Devices	436	5.88	22	4.15	0.80
.	.	. . .	.	.	.	.	.
.	.	. . .	.	.	.	.	.
M	14	Repair Electronic Governors	444	6.27	22	4.15	0.48

Table 5. New CODAP Programs for Analyzing and Displaying Task-Oriented Data

TSKFAC - Adds factor weight vectors to JOB DESCRIPTION file.

FACSTD - Generates file of rater background information and raw ratings.

FACSPC - Computes, displays, and saves mean factor ratings for identified groups of raters.

COMGEN - Generates and saves factors and factor composites through user-specified operations.

TSKCOR - Extracts vectors from JOB DESCRIPTION file, computes inter-correlations and user-specified regression equations.

PREFAC - Applies equation weights developed by TSKCOR, or CORREG, and produces predictor score vector.

TSKCAT - Generates user-specified vector of 1's and 0's to identify tasks of interest. Provides input to TSKFAC.

FACSUM - Flexible program for selecting, sorting, and printing task-factor information. Computes new vectors based on differences between vectors, maximums or minimums of vector sets, cumulative percentages, and categories of tasks developed by six arithmetic operators.

The first program is called TSKFAC. It is used to create new records on the JOB DESCRIPTION file which represent values for task factors. Decks containing factor values may be keypunched. However, such decks are normally generated by other CODAP programs, such as AVALUE, AVGPCT, REXALL, TSKGRP, or TSKNDX. The TSKFAC program produces Request cards, which can serve as control cards in AUTOJT, MTXPRT, FACSUM, and PRIJOB. Several factors can be written on the JOB DESCRIPTION file with a single execution of TSKFAC.

The second program, labeled FACSTD, serves the same function as the INPSTD does for job survey raw data in the present CODAP system, except it is used to create a Rater History file instead of a worker history file. Input to this program is the SETCHKed raw data file, which contains rating and history information. Input also includes Duty/Task title cards and History variable definitions. In the UNIVAC 1108 version, FACSTD will accept 20,000 raters (a highly unlikely encounter), 1700 task ratings, 1000 history variables, and 26 duty variables.

FACSPC calculates and publishes raw and standardized factor means for special groups of raters whose membership is defined in terms of some combination of background or computed variables. The mean ratings may be reported in high to low sequence on mean values, or on the percent of members performing, or in duty-task sequence. An option is provided for writing the output on the REPORT and/or JOB DESCRIPTION files. In some ways, FACSPC parallels the JOBSPC program, except that it applies to factor ratings instead of job descriptions.

COMGEN is a new program which primarily is used to generate composite factors in accordance with user specifications. However, this program is very general-purpose, operating on factor data in somewhat the same manner as PROGEN currently operates on history and task data. The COMGEN program generates coding from user-supplied verbs and FORTRAN statements to perform various mathematical operations on vectors from the JOB DESCRIPTION file.

TSKCOR extracts up to 100 vectors from a CODAP JOB DESCRIPTION file and computes matrices of intercorrelations as well as user-specified regression problems. The correlation matrices may be saved on tape for future use by CORREG.

PREFAC accepts equation weights developed by TSKCOR or CORREG and produces a predicted score deck which is ready for input by TSKFAC.

TSKCAT is a simple program which translates card decks containing task numbers to card decks with the identified tasks represented as "1's" in a NTASK--long string (vector) of 1's and 0's. The output deck is designed for input into TSKFAC and other programs for later use as an operator vector. For example, it can be used to suppress information in a FACSUM report.

FACSUM is the most flexible program in the new task-oriented package. This program allows the user to print any of the vectors on the JOB DESCRIPTION file. However, during the same run, the user can generate and publish new vectors representing differences between two vectors; the maximums or minimums of sets of vectors; cumulative percentages; or categories of tasks using any one of six arithmetic operators. Inclusion of sequence numbering, blank columns, sorting, and print suppression are under user control.

Now that the new programs have been enumerated, let's go back through and add a few details. First, it should be recognized that all of the new programs are now an integral part of the CODAP system. They interact with each other and with CODAP programs in the previous version. FACSUM can extract, manipulate, and display any vectors on the JOB DESCRIPTION file, including the "percent performing," "time spent by those performing," and "time spent by total group" vectors associated with any previously computed job description. Task-factor vectors generated by five previously available CODAP programs can be written on the JOB DESCRIPTION file by the TSKFAC program. Task factors placed on the JOB DESCRIPTION file can be manipulated by five previously available CODAP programs. Output from the TSKCOR program can be addressed by the CORREG program. Reports generated by three of the new programs can be written on the CODAP REPORT file and later extracted and published using request cards.

Now let's go through a semi-hypothetical example to illustrate how some of the task-oriented CODAP programs might be used in a study. To provide a degree of linkage with the paper being presented by Lt Col Mead, I will describe a study in which the goal is to establish training priorities for police patrolman tasks in the civilian sector. We will assume that supervisory ratings have been collected on three task factors which have relevance for our decision. These are (a) consequences of inadequate performance of each task, (b) task delay tolerance, and (c) task learning difficulty.

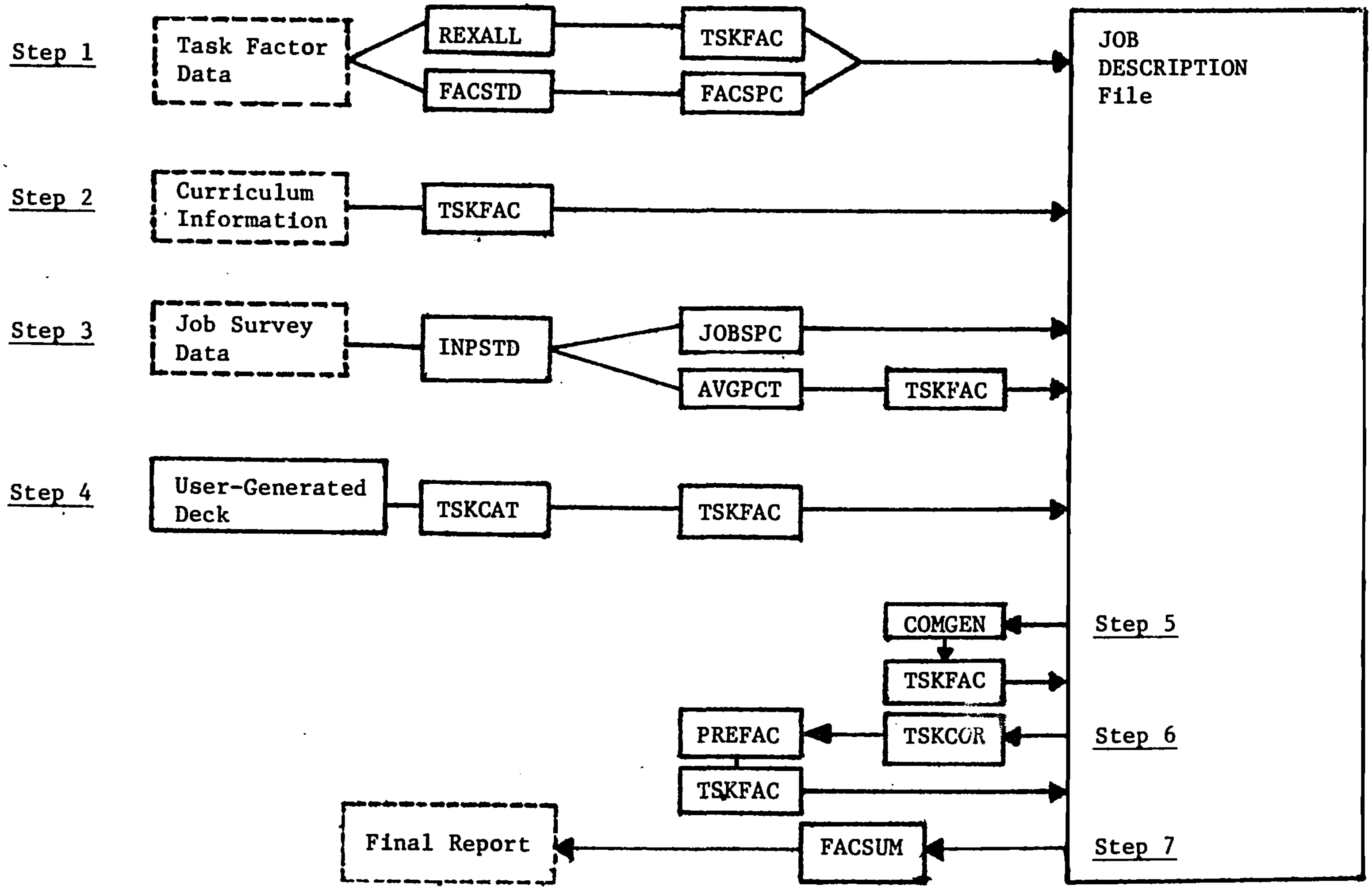
A job inventory containing all of the tasks performed by individuals in the police patrolman career ladder has been administered to a large sample of workers. Instructors in the training academy have indicated those tasks which are currently being taught. Our goal is to recommend for consideration certain modifications in course content and to support these recommendations. Figure 1 presents a simplified schematic of one possible route to the solution. In Step 1, we will process the supervisory task ratings on the consequence, delay tolerance, and difficulty factors. This could be accomplished by going through the REXALL and TSKFAC programs. The mean task ratings for all three factors could be written on the JOB DESCRIPTION file with a single TSKFAC run. An alternative route would be to go through the FACSTD and FACSPC programs, but this would not provide the basis for identifying raters for elimination or computing reliability coefficients. However, the latter route would enable the investigator to compare ratings provided by subgroups. For example, he could compute the mean ratings provided by police lieutenants and sergeants using FACSPC, and then display differences in their perspectives using the FACSUM program. A new program, called REXSPC, is being written to facilitate communication between the FACSTD and REXALL programs. Be that as it may, either of the two routes shown on the schematic can be used to get the mean factor ratings for the total sample onto the JOB DESCRIPTION file.

Since all of the information we have on the course curriculum is a vector of 1's and 0's indicating whether each task is or is not in the present course, we simply punch this information into a card deck and enter it onto the JOB DESCRIPTION file using the TSKFAC program. This is represented in the schematic as Step 2.

In Step 3, we generate two vectors from the job survey data file. First, we compute a job description using the JOBSPC program for patrolmen who have been on the job less than 12 months since graduation from the academy. The "percent performing" vector in this description can be used to estimate the probability that academy graduates will be required to perform each task during their first year on the job, which is one of the most valuable factors for making curriculum decisions.

The second factor generated from the survey data file is designed to identify tasks which are not relevant for consideration for inclusion in the academy course, since they are inappropriate for assignment to privates and corporals. In this case, we apply the AVGPCT program to the grade variable. This program computes the sum of the cross-products of percent

Figure 1. Schematic of Example Problem Utilizing Task Oriented CODAP Results



product of many years of curriculum judgments. Our goal o capture the systematic variance in this judgment process regression equation which gives consideration to the ors previously enumerated. We then will apply the equa- to every task in the occupational area. By this method, ay be able to identify tasks not being trained which would rained if the judgment process had been systematic. In same manner, we may be able to identify tasks being ned which the policy equation suggests should not be ned. The rationale for this approach will be discussed in detail when we view the final reports generated in Step 7.

It was hypothesized that certain of the task rating ors might be related to the criterion in a non-linear ion. In order to test this hypothesis, we needed to re certain vectors for introduction into the regression tions. This was accomplished in Step 5 by the COMGEN ram.

At this point, we have available on the JOB DESCRIPTION all variables required for computing the final regres- equation in Step 6. This is accomplished using the OR program. In the same run, the computed regression hts are applied to predictors using the PREFAC program, this vector is written onto the JOB DESCRIPTION file using TSKFAC program.

The final results are displayed by reports generated in 7 using the FACSUM program. I'm sorry that vugraphs are restricted in space to permit a full display of FACSUM , but I will present information extracted from one run ables 6 and 7. First let's look at Table 6.

In the first column, the FACSUM program has listed the

Table 6. Example FACSUM Output

Task Title	Tng Priority	In/Out Course	% Performing	Dly Tolerance	Consequences	Difficulty	Cons(-)Dif	Avg Grade
Conduct Frisk Search	1.14	1	85	1.6	7.3	4.3	3.0	3.5
Subdue Subject Resisting Arrest	1.10	1	69	1.1	7.6	5.2	2.4	3.5
Engage in High-Speed Pursuit Driving	1.09	0	69	1.1	7.0	5.9	1.1	3.5
Fill Out Theft Report Forms	1.00	1	81	5.6	5.5	4.6	.9	3.5
Advise Suspects of Their Legal Rights	1.00	1	68	2.1	6.6	3.6	3.0	3.6
Issue Moving Traffic Citations	.91	1	89	2.4	5.8	3.6	2.2	3.5
Check Autos Against Stolen Car List	.89	1	87	3.2	4.5	2.8	1.7	3.5
Issue Parking Citations	.78	1	51	4.7	3.1	2.8	.3	3.4
Take Statements in Criminal Cases	.59	0	32	4.1	6.3	6.1	.2	3.7
Deliver Departmental Mail	.41	0	29	6.5	2.5	2.6	-.0	3.7
Show Mug Shots to Witnesses	.40	1	16	5.6	5.5	4.2	1.2	4.1
Investigate Repossession Complaints	.30	0	24	6.7	2.5	4.5	-2.0	3.9
Punch Cards for Data Processing System	.22	0	1	6.8	4.5	5.9	-1.4	2.4
Photograph Lineup	.14	0	3	4.6	4.7	4.9	-.1	4.1
Train/Care for Dogs (Canine Unit)	.09	0	2	6.4	4.5	7.0	-2.4	3.2
Conduct Airborne Traffic Surveillance	.08	1	10	3.2	2.6	5.5	-2.9	3.4
Inspect Men at Roll Call	.04	0	3	3.8	4.0	3.5	.4	4.7
Prepare and Present Speeches	-.02	1	7	7.8	3.0	5.9	-2.9	3.9

titles for the eighteen tasks selected for the example using the operator variable generated in Step 4 by the TSKCAT program. Column 2 displays the training priority indexes generated by the TSKCOR and PREFAC programs in Step 6. Notice that all information in Table 6 has been ordered from high to low on the training priority index. The FACSUM program provides the investigator with the option of ordering information from low to high or high to low on any vector--or to list data in duty/task order.

Column 3 identifies tasks currently being taught in the academy by a value of "1," and tasks which are not being taught by a value of "0." This vector was introduced onto the JOB DESCRIPTION file in Step 2 using the TSKFAC program.

Column 4 indicates the percent of patrolmen who have been on the job 12 months or less performing each task. This was generated in Step 3 using the JOBSPC program.

Columns 5, 6, and 7 present the mean task-factor ratings on Delay Tolerance, Consequences of Inadequate Performance, and Learning Difficulty. These mean values were computed in Step 1 using the REXALL program, and were introduced onto the JOB DESCRIPTION file using the TSKFAC program.

Column 8 reports the differences between the values in Columns 6 and 7, and has been included in the table to illustrate one of the many capabilities of the FACSUM program. High positive values in this vector identify tasks which are easy to learn, but for which the consequences of inadequate performance are severe. Examples are "advising suspects of their legal rights" and "conducting frisk searches." High negative values on the vector identify difficult tasks with minor consequences for inadequate performance, examples being "presenting speeches" and "conducting airborne traffic surveillance."

Column 9 presents the average grade values computed by the AVGPCT program in Step 3. It reveals that "Inspect Men at Roll Call" is a sergeant or lieutenant task, and not appropriate for inclusion in the academy entry-level course.

Now let's go back and look at Column 3, which indicates whether each task is or is not included in the present course. Remember that the tasks are ordered in terms of values or the Training Priority Index. Notice that training is already being given on seven of the top eight tasks. The one exception is the third task listed: "Engage in High-Speed Pursuit Driving." No training is presently included in the course on this task. The equation suggests that training on this task should he introduced if curriculum designers are to be systematic in their decision process. The factor data reveal that 69% of graduates encounter this task during the first year after graduation. It has the lowest Task Delay Tolerance among all of the tasks on this chart. This means that when the task is encountered, performance must begin immediately, or the opportunity to perform it rapidly perishes. There is no time to read a manual or consult a friend. The consequences of inadequate performance are severe. I'm not sure of the exact values, but I recall reading in a recent newspaper article that about 50% of high-speed chases end with an accident, and that a significant number end with a fatality. Data on the chart show the task to be difficult to learn, and indicate that it is appropriate for consideration for inclusion in the academy course.

Now let me qualify this statement. The equation actually reflects the need for entry-level training on each task. It does not address the question as to where that training should be given. It is possible that this particular task could be taught more economically in the job setting. Be that as it may, the equation does reflect a need for training.

Looking at the bottom of the chart, we find that training is currently being given on two of the bottom seven tasks. The bottom task, "Prepare and Present Speeches," is performed by only seven percent of academy graduates during their first year on the job. While the task is rated as being difficult to learn, performers are seldom under time pressure (Task Delay Tolerance = 7.8), and supervisors rate the consequences of inadequate performance as being minor. For somewhat the same reasons, the equation raises questions concerning the training being given on "Conduct Airborne Traffic Surveillance," which is performed by only 10% of the graduates.

Table 7 presents an extract of another FACSUM output obtained in the same CODAP run. In this case, a difference score was generated between the observed and predicted training priorities, and the tasks are ordered on that vector. In accordance with user specifications, those tasks with near-equal observed and predicted priorities were suppressed from the printout, leaving only those tasks for which training decisions are required.

I'm sure that you appreciate the impossibility of adequately describing the flexibility and utility of the new task-oriented CODAP programs in the few minutes available to me at this conference.

Table 7. Recommendations for Possible Course Change

Task Title	Tng Priority	In/Out Course	Difference	% Performing	Dly Tolerance	Consequences	Difficulty	Avg Grade
Engage in High-Speed Pursuit Driving	1.090	0	1.090	69	1.1	7.0	5.9	3.5
Take Statements in Criminal Cases	.592	0	.592	32	4.1	6.3	6.1	3.7

Tasks Omitted for Which:
Difference: TSKF06(D) - TFAK01(F) is Less Than .500

Task Title	Tng Priority	In/Out Course	Difference	% Performing	Dly Tolerance	Consequences	Difficulty	Avg Grade
Show Mug Shots to Witnesses	.400	1	-.600	16	5.6	5.5	4.2	4.1
Conduct Airborne Traffic Surveillance	.088	1	-.912	10	3.2	2.6	5.5	3.4
Prepare and Present Speeches	-.028	1	-1.028	7	7.8	3.0	5.9	3.9