1978

A Simple Experiment in Top-Down Design

Douglas E. Comer
Purdue University, comer@cs.purdue.edu

M. H. Halstead

Report Number:
78-292

A SIMPLE EXPERIMENT IN TOP-DOWN DESIGN

CSD-TR-292

Douglas Comer

M.H. Halstead

Computer Science Department
Purdue University
West Lafayette, Indiana 47907

November 1978

A SIMPLE EXPERIMENT IN TOP-DOWN DESIGN

Douglas Comer and M.H. Halstead

Keywords and Phrases: Software Science, Top-Down Design, Technical Writing, Modularity

Abstract:

In this paper we: 1) discuss the need for quantitatively reproducible experiments in the study of Top-Down design; 2) propose the design and writing of tutorial papers as a suitably general and inexpensive vehicle; 3) suggest the software science parameters as appropriate metrics; 4) report two experiments validating the use of these metrics on outlines and prose; and 5) demonstrate that the experiments tended toward the same optimal modularity.

The last point appears to offer a quantitative approach to the estimation of the total length or volume (and the mental effort required to produce it) from an early stage of the Top-Down design process. If results of these experiments are validated elsewhere, then they will provide basic guidelines for the design process.

Introduction:

The concept of "Top-Down" design of programming projects [3,7,10] is by now well known and gaining acceptance [9,11]. Essentially, Top-Down design begins with a statement of what is to be done and proceeds by dividing the task into smaller, more manageable subtasks. Each subtask is further divided until the smallest subdivisions correspond directly to constructs in the source language. Moreover, careful documentation of the process enables systematic "backing up" to correct design errors. Only those parts of the design affected by a change need to be reconsidered.

But the usefulness and applicability of the Top-Down design methodology has been evaluated qualitatively at best; quantitative measures tend to be clouded by the multiplicity of extraneous variables affecting any given application. Because a basic tenet of the scientific method requires the independent reproduction of a measurement experiment before any relationship it demonstrates can be accepted, and because of the large number of irrelevant, independent variables, knowledge of this important design methodology might be characterized as pre-scientific.

This is not to say, however, that the current state of affairs can be easily remedied. Even a simple approach to the measurement problem requires the solution of a number of sub-problems. For example, in order that a given measurement experiment be reproducible in an independent environment, it must not place an inordinate drain upon resources. Similarly, it must deal only with the basic properties pertinent to the design process which apply generally over a wide range of design problems. Furthermore, and perhaps more important, the quantities to be measured must be well-defined, unambiguous, and independent of the wide variation in human talent.

The metrics from Software Science [4] which have been applied to compu-

ter programs provide a candidate measurement scheme for evaluating a design methodology. Software science is concerned with the measurable properties of computer programs, and the relationships which hold among their mean or average values. The equations of Software Science predict average behavior of a set of programs and, therefore, imply measurement over sufficiently large samples. As a result, individual values from small programs exhibit considerable statistical variation. The measurements required for evaluation of the design process, however, meet the criteria given above: they are well-defined, unambiguous, and independent of human talent. Therefore, the Software Science metrics were chosen as a basis for evaluating the Top-Down design methodology.

It is clear that the design and implementation of a large software system would be most germane to evaluation of the Top-Down methodology. But it is equally clear that the available resources would preclude reproduction of the experiment elsewhere. Consequently, any evaluation obtained from an initial measurement would remain unconfirmed, and therefore, of minimal value. Instead, the vehicle chosen for measurement is something that can be designed and "implemented" anywhere -- a technical paper. When the class of paper is restricted to tutorials, so that the author can be said to "understand the problem statement" as soon as he has a tentative title, then the principles of Top-Down design can be applied, a system of outlining can be adopted, and the entire process can be studied quantitatively.

The next section explains in detail the design process selected for study, the conditions under which the experiments were conducted, and the measurements taken. Then, Section 3 summarizes the observed data, while Section 4 exhibits the relationships predicted by Software science, and provides a comparison between predicted and observed values.

2 Experimental Procedure:

Each author performed one measurement experiment using the same, controlled conditions. The first experiment involved a research tutorial [5], while the second experiment used a paper explaining a computer program [1]. In each case, the author knew his subject matter intimately before the design was begun, and therefore, "understood the problem statement" once a title had been formulated (i.e. the requisite research and background readings had already been completed).

In order to separate and measure the distinct parts of the design process, a strict "Top-Down" approach was followed. A title was formulated, an initial five-point outline was derived from the title, and an expanded outline was produced by further subdividing each point five ways. The result, a hierarchy of five-tuples, reflected the Top-Down methodology used to construct it. After smoothing the five-by-five expanded outline, the author formulated his draft version of the paper using it.

With rigid divisions between each phase of writing, the design process could be measured by accounting separately for the time required to write each of the following parts of the paper:

1. Title page
2. Basic five-point outline
3. Expanded, five-by-five outline
4. Refined outline
5. First draft of the paper
6. Abstract for the first draft

During each stage of the "design", and for each hand written page of the "implementation" phase, the author recorded the time required to the nearest minute or closer. After final completion of the paper, each timed unit of written material was analyzed, using the counting methods given in [4], Chapter 13. Basically, the method partitions the text into two categories,

call operators and operands in Software Science terminology. Operators, in addition to punctuation and font changes, include "function words" defined and listed by [8]. Function words encompass the articles, pronouns, prepositions, conjunctions, auxiliary verbs and named numbers; all other words count as operands.

3 Observed Data:

Detailed data, including the elapsed time (in minutes), and the observed counts of operators and operands, can be found in [6] and Appendix. For convenience, pertinent values have been summarized in Tables 1 and 2 using:

η_1' to denote the number of unique operators,

η_2' to denote the number of unique operands,

N_1 to denote the total operator occurrences, and

N_2 to denote the total operand occurences.

Because a completely mechanical method of counting was used, each unique character string was recorded separately. For example, all five of the words "use", "user", "users", "uses", and "using" made a contribution to η_2'. Consequently, when a value of η_1 or η_2 comparable to those for computer programs is needed, the observed values of η_1' and η_2' must be reduced. This can be done in either of two ways. Each synonymous usage, case change, number change, etc, may be identified and removed subjectively, or the relation $\eta=k\eta'$ with $k = 2/5$ given in [4] can be used. Because it is both objective and much easier, the second method will be used.

The values of total usage are unaffected by this problem, hence there is no difference in N between computer programs and English prose.

4 Testing the Application of Software Metrics to the Data:

Before discussing the design hypothesis, it would provide some degree of confidence if it were found that the data from both experiments are comparable

with similar data from computer programs. This can easily be accomplished by testing the vocabulary-length relation. According to [4], an estimate of the program length, $N=N_1 + N_2$, can be obtained from

$$\hat{N} = \eta_1 \log_2\eta_1 + \eta_2\log_2\eta_2 \qquad (1)$$

Values of $\hat{N}$ compared with N for the draft pages of each experiment have been calculated from Tables 1 and 2, using $\eta_1 = 0.4\eta_1'$ and $\eta_2 = 0.4\eta_2'$ as noted earlier. The results are given in Table 3.

For both experiments the relative errors are small, and deviate from zero by less than their standard deviations. This suggests that the software metrics and equations apply to the prose generated during this experiment.

5 A Quantitative Hypothesis

It seems clear that the resulting prose generated by using a Top-Down design must exhibit some degree of modularity, and it might even be expected that this modularity would, on average, conform to an optimum in one sense at least. In discussing modularity in computer programs, [4] suggested that the chunking concept of psychology provided an approach to the quantification of the module most readily suited to the human brain. According to that concept, the high speed memory portion of the brain handles five chunks simultaneously. Assuming that a human has the ability to handle five inputs and the ability to generate one output from them, and equating these chunks to the software metric η_2^* which is defined as the count of conceptually unique input/output parameters that the ideal module should have

$$\eta_2^* = 6$$

The potential volume of the ideal module is therefore

$$V^* = \eta^*\log_2\eta^* = (2+\eta_2^*)\log_2(2+\eta_2^*) \qquad (2)$$

or

$$V_M^* = 24$$

where equation (2), and all of the software equations required later can be found in [4]. Now, the actual volume V is related to the potential volume via

$$(V^*)^2 = \lambda V \tag{3}$$

where λ is the language level.

Strictly speaking, λ should be obtained as

$$\lambda = L^2 V \tag{4}$$

where the implementation level L is defined as

$$L = V^*/V \tag{5}$$

but because the potential volume is seldom observed directly, the approximation

$$\hat{L} = \frac{2}{\eta_1} \frac{\eta_2}{N_2} \tag{6}$$

is usually used to obtain the estimate

$$\hat{\lambda} = (\hat{L})^2 V \tag{7}$$

Because $\hat{L}$ is only an approximation to L, $\hat{\lambda}$ differs from λ. While the two values have sometimes been confused in previous work, the distinction is important here, where λ=1, and $\hat{\lambda}$ may be calculated from η_1 and η_2 as obtained below. From equation (3), it follows that for the ideal module, the volume is

$$V_M = (V_M^*)^2 = 576 \tag{8}$$

From any volume it is possible to calculate the length, provided the value of η_2^* is known. The volume is defined as

$$V = N \log_2 \eta \tag{9}$$

which can be expanded to

$$V = (\eta_1 \log_2 \eta_1 + \eta_2 \log_2 \eta_2) \log_2 (\eta_1 + \eta_2) \tag{10}$$

and the relation between η_1 and η_2 is determined by η_2^* according to

$$\eta_2 = A\, \eta_1 + B \tag{11}$$

where

$$A = \frac{\eta_2^* \log_2 (\ ^*_2/2)}{2 + \eta_2^*} \tag{12}$$

and

$$B = \eta_2^* - 2 \cdot A \qquad (13)$$

Solving this set of equations with V=576 and η_2^*=6 gives not only the length, but the values of η_1 and η_2 as well. In addition the basic parameters N_1 and N_2 can be obtained directly from N, provdied that the relationship between them is known. While N_1 and N_2 are frequently assumed to be equal [4], the weight of experimental evidence indicates that N_1 is approximately one-third larger than N_2, or $3N_1 = 4N_2$. Continuing to use the subscript M to indicate the value for the ideal module, we have

$$\eta_{1,M} = 12.07$$

$$\eta_{2,M} = 17.96$$

$$\eta_M = 30.03$$

$$\eta'_{1,M} = 30.2$$

$$\eta'_{2,M} = 44.9$$

$$\eta'_M = 75.1$$

$$N_{1,M} = 67.2$$

$$N_{2,M} = 50.4$$

$$N_M = 117.6$$

From these values we can then obtain

$$\hat{L}_M = 0.059$$

$$\hat{\lambda}_M = 2.01$$

$$\hat{V}^*_M = 34.03$$

and from the effort equation

$$E_M = (V^*)^3/\lambda^2 = 24^3 = 13{,}824 \text{ e.m.d.} \qquad (14)$$

Using the Stroud Number S=18 elementary mental discriminations (e.m.d.) per second, this gives

$$T_M = E_M/S = 768 \text{ seconds} \qquad (15)$$

At this point we have a complete quantitative profile of the ideal module

for English prose, but we have not yet introduced any of the experimental conditions (other than that it be conducted in English).

But if we simply assume that the design strategy of writing the draft version from a five by five point outline resulted in the elucidation of 5x5 or 25 conceptually unique concepts, then for both of the papers we should expect that $\eta_2^*=25$. The method used in obtaining N_M for $\eta_2^* = 6$ can then be repeated for $\eta_2^*=25$, to yield a total length of N=2034. Because the total length must be the sum of its parts, the number of modules to be expected is simply the ceiling of the quotient

$$M = N/N_M = 2034/117.6 = 17.3$$

or

$$M = 18$$

Returning to Table 3, it can be verified that for both papers 18 clocked intervals were devoted to the writing. As evidence that this was not completely fortuitous, it is interesting to examine the total writing times. If the writing was done in modular fashion, then we should expect

$$T = MT_M = 13{,}824 \text{ seconds} - 230.4 \text{ minutes}$$

For Experiment 1, the observed total was 299 minutes, and for Experiment 2 it was 230.75 minutes. On the other hand, had each paper been written in a completely unmodularized fashion, then according to equation (15), some 2000 minutes or 33 hours would have been required.

Consequently, it appears safe to say that each of the 18 measured parts of each draft represents an approach to the ideal module, and they will be examined as such in the next section.

6 Agreement between Observations and Hypothesis:

As noted earlier, more than a dozen interesting properties can be calculated for an ideal module. From direct observation, and using only the

entries in Tables 1 and 2, these same properties can be calculated for each module for each experiment. The results are shown in Table 4, where the experimental values are the averages for the 18 modules. The relative errors were calculated by subtracting the predicted value from the observed mean and dividing by the observed mean.

In examining Table 4, it can be noted that in all cases the theoretical values fall within the experimental errors of the observations. Consequently, they give no reason to reject the hypothesis that both authors independently tended to approach the same quantitative characteristics when they developed their tutorial type papers with the same Top-Down design.

7 Conclusions:

The reasonable degree of agreement observed in the measurements suggests several tentative conclusions. First, it indicates that the metrics and equations of software science provide an appropriate avenue for quantitative study of some important and hither to unmeasured aspects of the design process.

Second, and much less clearly, these initial experiments suggest that the Top-Down design approach may have contributed significantly to reducing the time required from a theoretical value of 33 hours to the similarly theoretical modularized value of 3.84 hours, by an expenditure of from 0.3 to 2.9 hours in the earlier stages of the design.

Third, the results indicate, again most tentatively, that a method exists for predicting at a very early stage what quantity of human effort or time will be required to implement a design, and the length or volume which that effort can be expected to produce. It should be emphasized, however, that the draft versions reported upon here were in no sense final, typist-ready manuscripts. On the contrary, that stage was reached after another complete iteration, suggestive of the debug runs of the computer programmer.

Finally, it is as true now as it was before the experiments, that until this or some alternative quantitative approach has been reproduced at more than one laboratory our knowledge must still be considered pre-scientific. Even then, of course, large numbers of experiments could be required to determine the extent to which such results could be extrapolated, or generalized toward a useful theory of design.

References

[1] Comer, D., "MAP: A Pascal Preprocessor for Large Program Development", Software-Practice and Experience (to appear).

[3] Dahl, O.J., E. Dijkstra and C. Hoare, Structured Programming, Academic Press, 1972.

[4] Halstead, M.H., Elements of Software Science, Elsevier North-Holland, 1977.

[5] Halstead, M.H., "Potential Impacts of Software Science on Life Cycle Management", in Software Phenomenology, U.S. Army Institute for Research in Management Information and Computer Science, August 1977, pages, 385-400.

[6] Halstead, M.H., "A Software Science Analysis of the Writing of a Technical Paper", Tech. Report No. 242, Computer Science Department, Purdue University, August 1977.

[7] McGowan, C., and J. Kelly, Top-Down Structured Programming Techniques, Petrocelli/Charter, 1975.

[8] Miller, G., E. Newman and E. Friedman, "Length Frequency Statistics of Written English", Information and Control 1 (1958), 370-389.

[9] Tausworthe, R., Standardized Development of Computer Software, Prentice Hall, 1977.

[10] Wirth, N., "Program Development by Stepwise Refinement", CACM 14:4 (April 1971), 221-227.

[11] Yeh, R., ed., Current Trends in Programming Methodologies, Prentice Hall, 1977.

Part	T(Minutes)	n_1'	n_2'	N_1	N_2
1. Title Page	1	4	5	10	6
2. 5 Point Outline	9.3	25	21	53	28
3. 5x5 Point Outline	11	35	31	96	49
	4.3	12	17	42	22
	6.7	20	19	39	23
	4	26	19	92	31
	2.2	9	8	26	8
4. Final Outline	21	24	53	160	69
	11	24	22	70	29
	21	27	32	47	39
	25	27	42	223	67
	17	20	31	80	52
	13	20	33	52	47
	16	22	36	62	43
	11	23	25	61	28
5. Draft Paper	28	37	43	83	54
	31	23	53	67	65
	10	36	45	79	63
	16	40	47	69	53
	22	44	46	122	72
	11	39	27	120	45
	20	31	49	73	57
	14	28	52	81	55
	21	37	47	84	52
	14	20	33	48	39
	15	29	37	56	43
	15	27	43	62	46
	12	30	30	63	31
	11	37	34	61	37
	17	39	35	91	52
	15	34	45	64	51
	17	31	45	81	58
	10	23	20	44	26
6. Draft Abstract	13	22	31	48	41

TABLE 1

Part	T(Seconds)	η_1'	η_2'	N_1	N_2
1. Title Page	175	8	20	30	20
2. 5 Point Outline	160	11	10	27	10
3. 5x5 Point Outline	855	29	67	117	85
4. Draft Paper					
Page 1	1095	35	57	87	73
2	1000	34	64	94	79
3	1330	38	58	112	75
4	790	35	51	86	65
5	840	36	53	85	63
6	1095	40	56	95	67
7	745	30	53	91	70
8	940	35	43	74	54
9	740	37	63	85	72
10	605	37	65	92	75
11	210	22	22	34	23
12	720	35	47	81	62
13	735	38	52	112	72
14	1160	47	57	98	66
15	945	40	54	99	69
16	420	34	33	53	39
17	405	25	52	69	64
18	70	11	12	12	12

Experiment 1				Experiment 2		
Page	N	$\hat{N}$	$(N-\hat{N})/N$	N	$\hat{N}$	$(N-\hat{N})/N$
1	137	128	.065	160	156	.024
2	132	123	.069	173	171	.012
3	142	130	.081	187	165	.118
4	122	144	-.177	151	142	.059
5	194	150	.226	148	149	-.006
6	165	99	.401	162	164	-.015
7	130	129	.006	161	136	.153
8	136	130	.043	128	124	.032
9	136	137	-.008	157	175	-.114
10	87	73	.159	167	180	-.076
11	99	99	.005	57	55	.031
12	108	108	.003	143	133	.071
13	94	86	.085	184	151	.181
14	98	109	-.110	164	182	-.112
15	143	115	.195	168	160	.049
16	115	126	-.098	92	100	-.091
17	139	120	.136	133	124	.065
18	70	53	.236	24	20	.156
Mean	125	114	.073	142	138	.030
S.D.	30	25	.139	43	43	.089

TABLE 2

Property	Theory	Experiment 1				Experiment 2			
		Mean			R. E.	Mean			R. E.
M	18	18			0.000	18			0.000
η_1'	30.2	32.5	$\pm$	6.7	0.071	33.8	$\pm$	7.9	0.107
η_2'	44.9	40.6	$\pm$	9.1	-0.106	49.6	$\pm$	14.1	0.095
η'	75.1	73.1	$\pm$	12.0	-0.027	83.4	$\pm$	20.8	0.100
N_1	67.2	74.9	$\pm$	21.0	+0.103	81.1	$\pm$	25.6	0.171
N_2	50.4	49.9	$\pm$	11.8	-0.010	61.1	$\pm$	18.3	0.175
N	117.6	124.8	$\pm$	29.6	0.058	142.2	$\pm$	43.3	0.173
V	576.	611.	$\pm$	165.	0.057	730.	$\pm$	247.	0.211
$\hat{L}$	.059	.053	$\pm$	.015	-0.113	.055	$\pm$	.034	-0.073
$\hat{V}^*$	30.0	30.8	$\pm$	7.1	0.026	34.0	$\pm$	8.2	0.118
$\hat{\lambda}$	2.01	1.69	$\pm$	0.76	-0.189	1.71	$\pm$	0.45	-0.175
T(sec.)	768.	997.	$\pm$	354.	0.230	769.	$\pm$	333.	0.001

TABLE 4

APPENDIX

Operator and operand counts of Comer tutorial.

Title Page

8:45:00 to 8:47:55; T = 175 sec = 2.92 min

OPERATORS			OPERANDS	
1.	UC	20	1. AID	1
2.	#	3	2. COMER	1
3.	.	2	3. COMPUTER	1
4.	,	1	4. DEPARTMENT	1
5.	:	1	5. DEVELOPMENT	1
6.	A	1	6. DOUGLAS	1
7.	IN	1	7. INDIANA	1
8.	TO	1	8. LAFAYETTE	1
		30	9. LARGE	1
			10. MAP	1
			11. MARCH	1
			12. PASCAL	1
			13. PREPROCESSOR	1
			14. PURDUE	1
			15. SCALE	1
			16. SCIENCE	1
			17. SOFTWARE	1
			18. UNIVERSITY	1
			19. W	1
			20. 1978	1
				20

Outline (Five Point)

8:48:05 to 8:50:45; T = 160 sec - 2.67 min

OPERATORS

1.	UC	12
2.	.	5
3.	1	1
4.	2	1
5.	3	1
6.	4	1
7.	5	1
8.	A	1
9.	IN	1
10.	OF	2
11.	TO	1
		27

OPERANDS

1.	ADVANTAGES	1
2.	AID	1
3.	CONCLUSIONS	1
4.	DEVELOPMENT	1
5.	FEATURES	1
6.	INTRODUCTION	1
7.	MAP	1
8.	OVERVIEW	1
9.	PREPROCESSOR	1
10.	SOFTWARE	1
		10

Outline (5 x 5)

8:50:55 to 8:59:30 515
9:28:35 to 9:34:15 340

T = 855 sec = 14.25 min

OPERATORS

1.	UC	29
2.	.	22
3.	⧺	4
4.	()	4
5.	,	3
6.	:	2
7.	-	1
8.	/	3
9.	;	1
10.	1	5
11.	2	5
12.	3	5
13.	4	5
14.	5	4
15.	Λ	3
16.	AS	1
17.	IN	3
18.	IS	1
19.	NO	1
20.	OF	5
21.	TO	1
22.	THE	1
23.	LESS	1
24.	WITH	1
25.	STILL	1
26.	AMONG	1
27.	OTHER	1
28.	WHICH	1
29.	ACROSS	2
		117

OPERANDS

1.	ADVANTAGES	1	34.	INTACTS	1
2.	AID	1	35.	INTRODUCTION	1
3.	BUILTIN	1	36.	LANGUAGE	1
4.	CLEARLY	1	37.	MACHINES	1
5.	CLOSE	1	38.	MACRO	2
6.	COMPILATION	2	39.	MACROS	5
7.	COMPILERS	1	40.	MAINTENANCE	1
8.	CONDITIONAL	2	41.	MAP	1
C9.	CONSIDERATIONS	2	42.	MINOR	1
10.	CONST	3	43.	MOTIVATE	1
11.	CONSTANTS	1	44.	NOTION	1
12.	CONSTRUCTION	1	45.	OVERVIEW	1
13.	DATA	1	46.	PACK	2
14.	DATE	1	47.	PARAMETER	1
15.	DESCRIPTIONS	1	48.	PASCAL	3
16.	DESIGN	1	49.	PORTABILITY	2
17.	DEVELOPMENT	1	50.	PREPROCESSOR	2
18.	DISADVANTAGES	1	51.	PROBLEMS	1
19.	DISJOINT	1	52.	PROCEDURES	1
20.	DOCUMENTED	1	53.	SCOPES	1
21.	EG	1	54.	SIMPLE	1
22.	EXAMPLES	1	55.	SHARING	1
23.	EXPANSION	1	56.	SOFTWARE	1
24.	EXPENSE	1	57.	SOURCE	1
25.	EXPR	1	58.	STRING	1
26.	EXPRESSIONS	2	59.	SUBSTITUTION	1
27.	EXTENSIONS	2	60.	SURVEY	1
28.	FEATURES	1	61.	TIME	1
29.	FILES	1	62.	TOKEN	1
30.	GENERIC	1	63.	USEFULNESS	1
31.	IE	1	64.	VS	1
32.	INCLUDED	1	65.	$\underline{<}$	1
33.	INCLUDES	2	66.	=>	1
			67.	< =	1
					85

Draft Page 1

9:36:45 to 9:55:00; T = 1095 sec = 18.25 min

OPERATORS

1.	VC	14	18.	IS	1
2.	ϕ	2	19.	OR	1
3.	.	4	20.	TO	3
4.	,	8	21.	AND	1
5.	'	1	22.	ARE	3
6.	:	1	23.	BUT	1
7.	___	1	24.	FOR	3
8.	-	4	25.	NOT	2
9.	" "	1	26.	THE	2
10.	[ ]	2	27.	INTO	2
11.	( )	1	28.	THAT	1
12.	1	1	29.	THIS	1
13.	A	9	30.	WERE	1
14.	AN	1	31.	WILL	1
15.	AS	3	32.	WHICH	2
16.	BY	1	33.	ALREADY	1
17.	IN	5	34.	SEVERAL	1
			35.	NONETHELESS	1
					87

OPERANDS

1.	ABBREVIATIONS	1
2.	APPEARED	1
3.	AVAILABLE	1
4.	BASICALLY	1
5.	COMBINE	1
6.	COMPILER	1
7.	CONVERT	1
8.	DESCRIBES	1
9.	DEVELOPMENT	1
10.	EARLY	1
11.	EXPAND	1
12.	EXTENSIVE	1
13.	FEATURES	1
14.	FILE	1
15.	FILES	1
16.	FORM	1
17.	FORTRAN	2
18.	HIGH	2
19.	INPUT	1
20.	INTRODUCTION	1
21.	LANGUAGE	1
22.	LANGUAGES	1
23.	LARGE	2
24.	LEVEL	2
25.	MACRO	3
26.	MCIL60	1
27.	MC ILROY	1
	MI	
28.	MISCELLANEOUS	1
29.	NEW	1
30.	NOTED	1
31.	NUMBERS	1
32.	PAPER	1
33.	PASCAL	2
34.	PERFORM	1
35.	PREPROCESSOR	3
36.	PREPROCESSORS	3
37.	PROGRAM	2
38.	PROVIDES	1
39.	REFORMAT	1
40.	SCALE	1
41.	SINGLE	1
42.	SOFTWARE	1
43.	SOURCE	5
44.	STANDARD	1
45.	STRUCTURED	1
46.	SUPPORTS	1
47.	SURVEY	1
48.	TRANSLATOR	1
49.	TRANSLATORS	1
50.	TYPICALLY	1
51.	USE	1
52.	USED	1
53.	USEFUL	1
54.	USERS	1
55.	VALID	1
56.	WIDE	1
57.	1960	1
		73

Draft Page 2

9:55:05 to 10:11:45; T = 1000 sec = 16.67 min

OPERATORS

1.	UC	12	18.	FOR	1
2.	.	9	19.	HOW	1
3.	,	8	20.	ITS	1
4.	¶	1	21.	THE	6
5.	[ ]	2	22.	ALSO	1
6.	( )	2	23.	EACH	1
7.	___	5	24.	HAVE	1
8.	A	13	25.	INTO	1
9.	AN	1	26.	MOST	1
10.	AS	1	27.	ONLY	1
11.	IN	3	28.	SUCH	1
12.	IS	2	29.	THEIR	1
13.	IT	1	30.	WERE	1
14.	OF	4	31.	WHERE	1
15.	TO	4	32.	WHILE	1
16.	AND	3	33.	WILL	1
17.	CAN	1	34.	WITH	1
					94

OPERANDS

1.	AID	1	33.	LINKED	1
2.	ALLOWS	1	34.	LOOSELY	1
3.	APPROACH	1	35.	MACHINE	2
4.	ASSEMBLERS	1	36.	MACRO	2
5.	BROW 65	1	37.	MACROS	1
6.	BUILT	1	38.	MAKING	1
7.	CHARACTER	3	39.	MANUFACTURERS	1
8.	CHARACTERS	1	40.	MAPPINGS	1
9.	CONSISTS	1	41.	MARK	1
10.	CORRESPONDING	1	42.	MERELY	1
11.	CORRESPONDS	1	43.	NAME	3
12.	DEFINING	1	44.	NEED	1
13.	DEFINITIONS	2	45.	NEWLINE	1
14.	DEFINITIONS	1	46.	PARAMETERS	1
15.	DESCRIBES	1	47.	PREFIXED	1
16.	DEVELOPMENT	1	48.	PREPROCESSOR	2
17.	DIFFERENT	1	49.	PREPROCESSORS	2
18.	DISTINGUISHED	2	50.	SEQUENCE	1
19.	EARLY	1	51.	SET	1
20.	END	1	52.	SINGLE	1
21.	EXAMPLE	1	53.	SOFTWARE	1
22.	FOLLOWING	1	54.	STRACHEY	1
23.	FORMAL	1	55.	STRA65	1
24.	GIVE	1	56.	STRING	1
25.	GPM	3	57.	SUBSTITUTE	1
26.	IDENTIFIER	1	58.	SUPPLIES	1
27.	INCLUDING	1	59..	SUPPLY	1
28.	INDEPENDENT	1	60..	TOOK	1
29.	INHERENTLY	1	61.	USER	2
30.	INPUT	1	62.	VALUE	3
31.	LANGUAGE	2	63.	VIEWED	1
32.	LINE	1			79

Draft Page 3

10:11:55 to 10:34;05; T = 1330 sec = 22.17 min

OPERATORS

1.	UC	20	20.	ARE	2
2.	.	9	21.	CAN	3
3.	,	5	22.	FOR	1
4.	/	3	23.	THE	13
5.		2	24.	WAS	1
6.	[ ]	2	25.	BEEN	1
7.	___	1	26.	BOTH	1
8.	;	1	27.	HAVE	1
9.	A	6	28.	LESS	1
10.	AN	1	29.	MANY	1
11.	BE	2	30.	MUCH	1
12.	BY	1	31.	SOME	1
13.	IN	8	32.	THAN	2
14.	IS	5	33.	THAT	1
15.	OF	5	34.	WHEN	1
16.	ON	1	35.	WITH	1
17.	OR	1	36.	WHICH	2
18.	TO	1	37.	RATHER	1
19.	AND	3	38.	ALTHOUGH	1
					113

OPERANDS

1.	ACTUAL	1	30.	MLJ	3
2.	ADDITION	1	31.	OPERATIONS	1
3.	ALTERNATIVE	1	32.	ORIENTATION	1
4.	APPLICATIONS	1	33.	OUTPUT	1
5.	BASIC	2	34.	PARAMETERS	1
6.	BASIS	1	35.	PERFORMED	1
7.	BROWN	1	36.	PREPROCESSING	1
8.	BROW67	1	37.	PREPROCESSOR	3
9.	C	3	38.	PREPROCESSORS	1
10.	CALL	1	39.	PROCESSOR	1
11.	CALLED	1	40.	PROGRAMMING	1
12.	CALLS	1	41.	PURPOSE	2
13.	CHANGES	1	42.	REPLACED	1
14.	CHARACTER	2	43.	RESTRICTED	1
15.	COMPILED	1	44.	SHARPLY	1
16.	CONTEXT	1	45.	SOURCE	1
17.	CONTRASTS	1	46.	SPECIAL	1
18.	DERJ78	1	47.	STRICTLY	1
19.	DESCRIPTION	1	48.	SUGGESTS	1
20.	DISADVANTAGE	1	49.	SUPPLY	1
21.	ENGLISH	1	50.	SYNTACTIC	1
22.	GENERAL	1	51.	SYNTAX	2
23.	GPM	3	52.	TOKEN	1
24.	IMMEDIATELY	1	53.	TOKENS	1
25.	LANGUAGE	2	54.	TYPE	1
26.	LIMITED	1	55.	UNIT	1
27.	MACRO	5	56.	UNITS	1
28.	MADE	1	57.	USER	1
29.	MAP	1	58.	WORDS	1
					75

Draft Page 4

1:23:00 to 1:44:50; T = 710. sec = 11.83 min

OPERATORS

1.	¶	1	19.	HAVE	1
2.	U.C.	15	20.	IN	2
3.	o	6	21.	ITS	1
4.	,	7	22.	MOST	1
5.	-	3	23.	OF	8
6.	[ ]	2	24.	OTHERS	1
7.	( )	3	25.	SEVERAL	1
8.	8	1	26.	THAT	1
9.	11	1	27.	THE	4
10.	A	6	28.	THESE	1
11.	AND	3	29.	THIS	1
12.	ARE	1	30.	TO	1
13.	AS	1	31.	WE	1
14.	BE	2	32.	WHILE	1
15.	BY	1	33.	WELL	1
16.	FIRST	1	34.	WILL	3
17.	FOR	1	35.	WITH	1
18.	HAS	1			86

OPERANDS

1.	AID	1	27.	LARGE	2
2.	ADDRESS	1	28.	MACRO	3
3.	ATTENTION	1	29.	MAP	2
4.	BRIEF	1	30.	MENTIONS	1
5.	CAREFULLY	1	31.	NEWS	1
6.	CHOSEN	1	32.	PAPER	1
7.	CONSIDER	1	33.	PARTICULAR	1
8.	CONSTRUCTION	1	34.	PASCAL	5
9.	DESCRIPTION	1	35.	PREPROCESSOR	1
10.	DEGIGN	2	36.	PREPROCESSORS	1
11.	DISCUSSED	1	37.	PROBLEMS	2
12.	DISCUSSION	1	38.	PROGRAMMING	1
13.	EFFECTIVENESS	1	39.	PROGRAMS	1
14.	EXTENSIONS	2	40.	PURPOSE	1
15.	FEATURES	2	41.	RECENTLY	1
16.	FOLLOWED	1	42.	SCALE	1
17.	GAINED	1	43.	SEE	1
18.	GENERAL	1	44.	SOFTWARE	1
19.	GIVEN	1	45.	STRUCTURES	1
20.	HOLT	1	46.	SUGGESTED	1
21.	IMPACT	1	47.	SYSTEMS	1
22.	IMPLEMENTATION	2	48.	VARIATIONS	1
23.	IMPLEMENTATIONS	1	49.	VARIETY	1
24.	IMPLEMENTING	1	50.	WIDE	1
25.	INTENDED	1	51.	WIR61	1
26.	LANGUAGE	2			65

Draft Page 5

1:45:00 to 1:59:00; T = 840 sec = 14.00 min

OPERATORS

1.	⊕	1	19.	HAVE	1
2.	UC	16	20.	IN	5
3.	.	7	21.	IS	2
4.	,	7	22.	IT	1
5.	:	1	23.	MAY	1
6.	2	1	24.	MOST	1
7.	A	1	25.	NOT	1
8.	AN	2	26.	OF	6
9.	AND	4	27.	ONLY	2
10.	ARE	1	28.	RATHER	1
11.	AS	1	29.	SINCE	1
12.	BE	1	30.	SO	1
13.	BEEN	1	31.	THAN	1
14.	DOES	1	32.	THE	5
15.	EACH	1	33.	THEM	1
16.	FEW	1	34.	TO	4
17.	FOR	1	35.	VERY	1
18.	FOUR	1	36.	WAS	1
					85

OPERANDS

1.	ADDED	1	28.	MACRO	1
2.	ADDITIONAL	1	29.	MADE	1
3.	ADDITIONS	1	30.	MAP	2
4.	APPLY	1	31.	MAX	1
5.	ATTEMPT	1	32.	MIN	1
6.	BASIC	1	33.	NAME	2
7.	CASE	1	34.	OPERATORS	1
8.	COMPILATION	1	35.	OVERVIEW	1
9.	CONCATENATION	1	36.	PASCAL	5
10.	CONDITIONAL	1	37.	PRESERVED	1
11.	CONST	2	38.	PREVIOUSLY	1
12.	CONSTANT	2	39.	PROHIBITED	1
13.	CONSTANTS	1	40.	PROVIDES	1
14.	CONTEXT	1	41.	RECOGNIZE	1
15.	COURSE	1	42.	REENFORCE	1
16.	DECLARATION	1	43.	RULES	1
17.	DECLARATIONS	1	44.	SCOPE	2
18.	DEFINED	1	45.	SOURCE	1
19.	EVALUATED	1	46.	STANDARD	1
20.	EVALUATION	1	47.	STRING	1
21.	EXAMPLE	1	48.	STRUCTURED	1
22.	EXPRESSION	2	49.	SUBSTITUTION	1
23.	EXPRESSIONS	1	50.	SUBVERT	1
24.	FILES	1	51.	SUPPLEMENT	1
25.	IMPORTANTLY	1	52.	TYPING	1
26.	INCLUDED	1	53.	USED	1
27.	LANGUAGE	1			63

Draft Page 6

1:59:05 to 2:16:40 and

2:51:50 to 2:52:30; T = 1095 sec = 18.25 min

OPERATORS

1.	¶	2	21.	FOR	1
2.	U.C.	14	22.	IN	4
3.	.	5	23.	IS	5
4.	,	4	24.	IT	3
5.	" "	1	25.	MANY	1
6.	:	1	26.	MAY	1
7.	/	1	27.	MOST	1
8.	( )	3	28.	NOT	1
9.	[ ]	1	29.	OF	7
10.	I	1	30.	ON	1
11.	A	5	31.	RATHER	1
12.	ALSO	1	32.	THAN	1
13.	AN	1	33.	THE	7
14.	ANOTHER	1	34.	THEY	1
15.	ARE	3	35.	THIS	1
16.	AS	2	36.	THUS	1
17.	BE	1	37.	TO	5
18.	BY	1	38.	WHICH	1
19.	CANNOT	1	39.	WHILE	1
20.	EVEN	1	40.	WITH	1
					95

OPERANDS

1.	ACTUAL	1	29.	ML	1
2.	ASPECT	1	30.	NOTION	1
3.	AVOIDS	1	31.	ORDER	1
4.	BASIC	1	32.	PARAMETER	4
5.	CALLED	1	33.	PASCAL	1
6.	CF	1	34.	PERFORM	1
7.	CHARACTER	1	35.	POWER	1
8.	COMPUTATIONAL	1	36.	PROCESSING	1
9.	CONFORM	1	37.	REFERENCE	1
10.	CONTAIN	1	38.	REFLECTED	1
11.	DEFFINITION	1	39.	RESTRICTED	1
12.	DEPENDING	1	40.	RESTRICTION	1
13.	DIFFERENT	1	41.	RESULTS	1
14.	EVALUATION	1	42.	SEVERELY	1
15.	EXAMPLE	1	43.	SIMPLICITY	1
16.	FACILITIED	1	44.	SIMULATE	1
17.	FORMAL	1	45.	SPECIAL	1
18.	GPM	1	46.	STRA65	1
19.	IMPLEMENTED	1	47.	SUBSTITUTION	1
20.	IMPOSSIBLE	1	48.	SYNTAX	1
21.	LANGUAGE	1	49.	SYSTEM	1
22.	LIMITED	1	50.	TOKEN	1
23.	LIMITS	1	51.	TRICKS	1
24.	MACHINES	1	52.	TURING	1
25.	MACRO	3	53.	UNIT	1
26.	MACROS	6	54.	USED	1
27.	MAKE	1	55.	USES	1
28.	MAP	2	56.	YIELD	1
					67

Draft Page 7

2:52:45 to 3:05:10; T = 745 sec = 12.42 min

OPERATORS

1.	¶	4	16.	DOES	1
2.	.	5	17.	FOR	3
3.	,	10	18.	HAVE	1
4.	:	1	19.	IF	2
5.	U.C.	9	20.	IN	2
6.	$	3	21.	IS	1
7.	( )	5	22.	IT	4
8.	___	5	23.	MAY	2
9.	" "	1	24.	NOT	2
10.	A	3	25.	OF	5
11.	ANOTHER	1	26.	THAT	3
12.	ARE	1	27.	THE	9
13.	AS	1	28.	TO	3
14.	AT	1	29.	WHILE	1
15.	BUT	1	30.	WITH	1
					91

OPERANDS

1.	ACTUAL	1	28.	LIST	1
2.	ADVANTAGE	1	29.	MACRO	5
3.	AVAILABLE	1	30.	MATCHES	1
4.	BOOL	1	31.	MULTIPLE	1
5.	BOOLEAN	1	32.	NAME	2
6.	C	1	33.	NAMES	1
7.	CALL	2	34.	NEW	1
8.	CODED	1	35.	PARAMETERS	1
9.	CODEIF	4	36.	PASCAL	1
10.	CODING	1	37.	PERSE	1
11.	COMPILATION	2	38.	POINT	1
12.	COMPOUND	1	39.	PREFERRED	1
13.	CONDITIONAL	2	40.	PROBLEM	1
14.	CONST	1	41.	PROVIDES	2
15.	CONTAIN	1	42.	REFERENCES	1
16.	COURSE	1	43.	SEEM	1
17.	DEFINE	3	44.	SIMILARLY	1
18.	DESIRABLE	1	45.	SOURCE	1
19.	ENDIF	1	46.	STATEMENT	2
20.	EXAMPLE	1	47.	STATEMENTS	1
21.	EXPRESSION	2	48.	SYNTAX	1
22.	FILES	1	49.	SYSTEM	2
23.	FORM	1	50.	TOKEN	1
24.	FORMALS	1	51.	TOKENS	1
25.	INCLUDE	1	52.	USED	1
26.	INCLUSION	1	53.	VALUE	1
27.	INTRODUCED	1			70

Draft Page 8

3:05:30 to 3:21:00 T = 940 sec = 15.67 min

OPERATORS

1.	¶	1	19.	DOES	1
2.	UC	7	20.	HAVE	1
3.	o	5	21.	IF	2
4.	,	4	22.	IN	4
5.	;	1	23.	IS	1
6.	" "	2	24.	NO	1
7.	[ ]	1	25.	NOT	1
8.	#	2	26.	OF	7
9.	A	1	27.	SO	1
10.	ALL	1	28.	THE	12
11.	AN	1	29.	THIS	1
12.	AND	1	30.	TO	2
13.	ANOTHER	1	31.	WELL	1
14.	ARE	2	32.	WHEN	1
15.	AS	2	33.	WHILE	1
16.	BEEN	1	34.	WITH	1
17.	BEING	1	35.	WOULD	1
18.	CANNOT	1			74

OPERANDS

1.	ACTIVE	1	23.	LANGUAGE	2
2.	ASSUMED	1	24.	LIST	1
3.	ANNOYING	1	25.	MACRO	3
4.	CALLS	1	26.	MACROS	3
5.	CONTAIN	1	27.	MAP	1
6.	CURRENTLY	1	28.	MATCHES	1
7.	DETAIL	1	29.	NEW	1
8.	ENDIF	1	30.	NOTIONS	1
9.	ENVIRONMENTS	2	31.	POTENTIAL	1
10.	EXITS	1	32.	PREFERABLE	1
11.	EXPANDED	1	33.	PREPROCESSING	1
12.	EXPANSIONS	1	34.	PREPROCESSOR	1
13.	FILE	3	35.	PREPROCESSORS	1
14.	FILES	2	36.	RESTRICTION	1
15.	FORM	1	37.	SEEMS	1
16.	IMPLEMENTATION	1	38.	SIMPLIFY	1
17.	INCLUDED	1	39.	STACK	3
18.	INCLUSION	1	40.	STRICTLY	1
19.	INTERFER	1	41.	TERMS	1
20.	INTERFERENCE	1	42.	TOKEN	1
21.	INVOLVES	1	43.	UPDATED	1
22.	KR78	1			54

Draft Page 9

3:21:10 to 3:33:30; T = 740 sec = 12.33 min

OPERATORS

1. ¶	2	19.	IN	3
2. U.C.	15	20.	IS	1
3. .	7	21.	MIGHT	1
4. ,	5	22.	MORE	1
5. -	2	23.	MOST	1
6. " "	1	24.	NOT	1
7. /	2	25.	OF	2
8. 3	1	26.	ON	1
9. III	1	27.	OWR	1
10. A	8	28.	SEVERAL	1
11. AN	1	29.	SOME	1
12. AND	2	30.	SOMETIMES	1
13. ARE	1	31.	SOMEWHAT	1
14. BE	1	32.	THAT	1
15. BY	1	33.	THE	5
16. EVEN	1	34.	TO	5
17. FIRST	1	35.	WHICH	1
18. FOR	2	36.	YET	1
				83

OPERAND

1.	ADDITION	1	33.	MESSAGE	1
2.	AID	1	34.	MULTIUSER	1
3.	BUILTIN	1	35.	NECESSARY	1
4.	BUG	1	36.	NUMBER	1
5.	CLAIM	1	37.	PAGE	1
6.	COMPILATION	1	38.	PASCAL	1
7.	COMPILE	1	39.	PHILOSOPHY	1
8.	COMPILER	2	40.	PLEASANT	1
9.	COMPILERS	1	41.	PRINTS	1
10.	CONSIDER	1	42.	PROGRAM	2
11.	CONST	1	43.	PROVIDES	1
12.	CONSTANTS	1	44.	PROVIDING	1
13.	CONTAINED	2	45.	REFLECT	1
14.	DATE	2	46.	REPORTS	1
15.	DECKS	1	47.	SIMPLE	1
16.	DESIGNED	1	48.	SINGLE	2
17.	DEVELOPMENT	2	49.	SOFTWARE	1
18.	ENVIRONMENT	1	50.	STAMP	2
19.	EXAMPLES	1	51.	SUPPORT	1
20.	EXPRESSIONS	1	52.	SYSTEM	1
21.	FACILITIES	1	53.	TIME	1
22.	FEATURES	1	54.	TOOL	1
23.	FOLLOW	1	55.	TRACE	1
24.	FOSTERS	1	56.	TYPICALLY	1
25.	HEADER	1	57.	USER	1
26.	HELP	1	58.	VERSION	2
27.	HELPFUL	1	59.	WRITING	1
28.	INDIVIDUAL	1	60.	WRITTEN	1
29.	LANGUAGES	1	61.	X	1
30.	LARGE	1	62.	10.2	1
31.	LIKE	2	63.	14	1
32.	MAP	2	64.	78	1
					74

Draft Page 10

3:33:35 to 3:43:40; T = 605 sec = 10.08 min

OPERATORS

1.	¶	3	21.	BY	1
2.	U.C.	13	22.	CAN	1
3.	.	6	23.	COULD	1
4.	,	9	24.	EACH	1
5.	:	1	25.	FROM	1
6.	;	1	26.	IN	4
7.	' '	2	27.	INTO	2
8.	___	1	28.	IS	1
9.	( )	2	29.	OF	5
10.	[ ]	2	30.	ON	1
11.	=	1	31.	ONE	1
12.	/	1	32.	OR	1
13.	I	1	33.	THE	10
14.	A	6	34.	THESE	1
15.	ALSO	1	35.	THUS	1
16.	ALWAYS	3	36.	TO	2
17.	AND	1	37.	UP	1
18.	ANOTHER	1	38.	WHEN	1
19.	AS	1	39.	YET	1
20.	BE	1		.	94

OPERANDS

1.	AUTOMATICALLY	1	33.	INCLUDES	1
2.	BORROWED	1	34.	INCLUDING	1
3.	C	1	35.	KEEP	1
4.	CAPTURED	1	36.	KERT68	1
5.	CHANGE	2	37.	LARGE	1
6.	CODE	1	38.	MAKING	1
7.	COMPILATION	1	39.	MAP	2
8.	COMPILE	1	40.	PL	1
9.	COMPILER	1	41.	PREDEFINED	1
10.	CONST	2	42.	PROCEDURES	1
11.	CONSTANTS	2	43.	PROGRAMMERS	1
12.	CORRECT	1	44.	PROJECT	1
13.	DATE	4	45.	PROVIDED	1
14.	DECLARATION	1	46.	PROVIDES	1
15.	DIVIDE	1	47.	REFLECT	1
16.	EASILY	2	48.	RUN	1
17.	EFFORT	1	49.	SEPARATE	1
18.	EXPRESSIONS	1	50.	SET	1
19.	FACILITY	1	51.	SOURCE	1
20.	FILE	2	52.	STAMP	1
21.	FILES	1	53.	STATEMENT	1
22.	FORGET	1	54.	TIME	1
23.	FORMULATED	1	55.	USEFUL	1
24.	FOUND	1	56.	USING	1
25.	GROUP	2	57.	USUALLY	1
26.	GROUPS	1	58.	VERS	1
27.	HEAD	1	59.	VERSION	1
28.	HEADING	1	60.	WORKING	1
29.	HEADINGS	1	61.	WORSE	1
30.	IBM	1	62.	WRITE	1
31.	IMBEDDED	1	63.	X	1
32.	INCLUDE	1			74

Draft Page 11

3:43:50 to 3:47:20; T = 210 sec = 3.50 min

OPERATORS			OPERANDS		
1.	UC	1	1.	AID	1
2.	°	3	2.	COMBINE	1
3.	,	2	3.	COMPILATION	1
4.	A	1	4.	COMPILERS	1
5.	ALL	1	5.	COPYING	1
6.	AN	1	6.	COST	1
7.	BE	2	7.	ELIMINATED	1
8.	CAN	1	8.	FILE	2
9.	FEW	1	9.	FILES	1
10.	FOR	1	10.	IMPLEMENTED	1
11.	IN	2	11.	INCLUDES	1
12.	INTO	1	12.	INCLUSION	1
13.	IS	3	13.	LARGE	1
14.	IT	2	14.	OBVIOUS	1
15.	MANY	1	15.	PROGRAM	1
16.	OF	1	16.	PROGRAMMING	1
17.	ONE	2	17.	SMALL	1
18.	ONTO	1	18.	SOURCE	1
19.	SO	1	19.	SUPPORT	1
20.	SUCH	2	20.	SURPRISING	1
21.	THAT	2	21.	SYSTEMS	1
22.	THE	2	22.	WAY	1
		34			23

Draft Page 12

10:29 to 10:41:00; T = 720 sec = 12.00 min

OPERATORS

1.	#	1	19.	IT	1
2.	UC	8	20.	NO	1
3.	.	6	21.	NOT	1
4.	,	8	22.	OF	3
5.	-	1	23.	ONE	1
6.	A	3	24.	ONLY	1
7.	ACROSS	1	25.	OTHER	1
8.	ARE	1	26.	OTHERS	1
9.	BE	3	27.	SAME	1
10.	BUT	1	28.	SEVERAL	2
11.	BY	1	29.	SINCE	1
12.	CAN	2	30.	THAT	1
13.	FOR	3	31.	THE	6
14.	FROM	1	32.	THEREFORE	1
15.	HAS	1	33.	THEIR	1
16.	IF	2	34.	TO	5
17.	IN	2	35.	WHEN	2
18.	IS	6			81

OPERANDS

1.	ACTIVATED	1	25.	OFF	2
2.	ADDED	1	26.	OVERHEAD	1
3.	ADDITION	1	27.	PROGRAM	2
4.	ADVANTAGES	1	28.	PROGRAMMERS	1
5.	BOOLEAN	1	29.	PROVIDED	1
6.	BUILDING	1	30.	READY	1
7.	CODE	3	31.	RUN	1
8.	CODEIF	2	32.	SCALE	1
9.	COMPILATION	2	33.	SET	1
10.	COMPILED	1	34.	SETS	1
11.	CONDITIONAL	2	35.	SINGLE	1
12.	DEBUG	3	36.	SOFTWARE	1
13.	DECK	2	37.	SOURCE	3
14.	DIFFERENT	2	38.	SWITCH	2
15.	EXAMPLE	1	39.	SYSTEM	1
16.	FACILITY	1	40.	TRANSPORTED	1
17.	GENERATE	1	41.	TIME	1
18.	INTRODUCES	1	42.	TURNED	1
19.	LARGE	1	43.	USE	1
20.	LEAVE	1	44.	USED	1
21.	MACHINES	1	45.	USEFUL	1
22.	MACRO	1	46.	VERSIONS	2
23.	MAP	1	47.	WILLING	1
24.	OBVIOUS	1			62

Draft Page 13

10:41:55 to 10:54:10; T = 735 sec = 12.25 min

OPERATORS

1.	⫽	2	20.	BE	3
2.	UC	13	21.	BECAUSE	1
3.	.	12	22.	BETWEEN	1
4.	,	3	23.	BY	1
5.	:	1	24.	EACH	1
6.	;	1	25.	FURTHER	1
7.	-	2	26.	HAVE	1
8.	()	1	27.	IS	1
9.	1	1	28.	LESS	1
10.	2	1	29.	NOT	2
11.	3	1	30.	OF	6
12.	4	1	31.	ON	1
13.	IV	1	32.	THAN	1
14.	5	1	33.	THE	15
15.	6	1	34.	TO	8
16.	A	8	35.	TWO	1
17.	AFTER	1	36.	WAS	1
18.	AND	6	37.	WERE	2
19.	ALREADY	1	38.	WOULD	6
					112

OPERANDS

1.	ADVANTAGES	1	27.	LATTER	1
2.	BLUR	1	28.	LOCAL	1
3.	CAREFULLY	1	29.	MACHINE	1
4.	CHANGES	1	30.	MACRO	1
5.	CHOSEN	1	31.	MAP	4
6.	COMPILER	8	32.	MODIFICATION	1
7.	COMPLETED	1	33.	MODIFICATIONS	1
8.	CONSIDERED	2	34.	MODIFIED	1
9.	DEFINITELY	1	35.	MODIFY	1
10.	DEVELOP	1	36.	NEW	2
11.	DISADVANTAGES	1	37.	PASCAL	2
12.	DISTINCTION	1	38.	POORLY	1
13.	DOCUMENTED	1	39.	PORT	1
14.	EASY	1	40.	PREPROCESSOR	3
15.	EXISTING	3	41.	PROCESSING	1
16.	EXTENSION	1	42.	PROVIDED	1
17.	EXTENSIONS	1	43.	REAPPLIED	1
18.	FILES	1	44.	RELEASE	1
19.	FLY	1	45.	REQUIRED	2
20.	HUGE	1	46.	SEPARATE	1
21.	IMPLEMENTATION	1	47.	SIZE	1
22.	IMPLEMENTATIONS	1	48.	SPECIFICATIONS	1
23.	INCLUDING	1	49.	STANDARD	1
24.	INCREASE	1	50.	TIME	2
25.	INDEPENDENT	2	51.	TOOL	1
26.	LANGUAGE	1	52.	UNDERSTAND	1
					72

Draft Page 14

10:54:15 to 11:13:35; T = 1160 sec = 19.33 min

OPERATORS

1.	#	2	25.	FROM	1
2.	UC	13	26.	IN	4
3.	.	7	27.	IS	1
4.	,	6	28.	IT	1
5.	;	1	29.	ITS	1
6.	'	1	30.	MOST	2
7.	[]	1	31.	NOT	1
8.	#	1	32.	OF	3
9.	4	1	33.	OR	1
10.	IV	1	34.	PERHAPS	1
11.	A	2	35.	QUITE	1
12.	ALMOST	1	36.	SAME	2
13.	ALREADY	1	37.	SEVERAL	1
14.	AMONG	1	38.	SINCE	1
15.	AN	1	39.	THAT	3
16.	AND	1	40.	THE	11
17.	ARE	1	41.	TO	4
18.	AS	1	42.	TWO	1
19.	AT	1	43.	WAS	3
20.	BETWEEN	1	44.	WELL	1
21.	BY	1	45.	WHERE	1
22.	DID	1	46.	WITH	2
23.	ENOUGH	1	47.	WOULD	2
24.	EVERYONE	1			98

OPERANDS

1.	ALPARD	1	30.	MAINTAIN	1
2.	APPARENT	1	31.	MAP	3
3.	ASSOCIATE	1	32.	MODEL	1
4.	ATTEMPT	1	33.	PARTICULAR	1
5.	AUTHOR	1	34.	PASCAL	4
6.	BASIC	1	35.	POTENTIAL	1
7.	CDCS	1	36.	PREPROCESSOR	1
8.	CLU	1	37.	PROBABLY	1
9.	COMPILER	1	38.	PROBLEM	1
10.	CONFUSE	1	39.	PROGRAMMERS	1
11.	CONSCIOUS	1	40.	PROVIDE	1
12.	DESIGNED	1	41.	REACTION	1
13.	DISADVANTAGES	1	42.	REASON	1
14.	DISTINCTION	1	43.	RESULT	1
15.	ENCAPSULATION	1	44.	SEEMS	1
16.	ENVIRONMENT	1	45.	SIMILAR	1
17.	ESPECIALLY	1	46.	SIMULTANEOUSLY	1
18.	EXTENDED	1	47.	SOLUTION	1
19.	EXTENDING	1	48.	STUDENTS	2
20.	FACILITY	1	49.	SUPPLIED	1
21.	FACILITIES	1	50.	SURPRISING	1
22.	FEAR	1	51.	TENDENCY	1
23.	FLEDGLING	1	52.	TIME	1
24.	FORGET	1	53.	UNDESIRABLE	1
25.	FORTRAN	2	54.	UNIVERSITY	1
26.	IMPLEMENTATION	1	55.	USERS	2
27.	INTERESTING	1	56.	VERSIONS	1
28.	LANGUAGE	2	57.	WANTED	1
29.	LEARN	1			66

Draft Page 15

11:13:40 to 11:29:25; T = 945 sec = 15.75 min

OPERATORS

1.	¢	1	21.	NOT	1
2.	UC	7	22.	OF	7
3.	.	3	23.	ONE	2
4.	,	11	24.	OTHER	1
5.	:	1	25.	OVER	1
6.	-	2	26.	OWN	1
7.	" "	1	27.	SOME	1
8.	()	1	28.	STILL	1
9.	A	4	29.	SUCH	1
10.	ACROSS	1	30.	THE	13
11.	AND	3	31.	THEM	1
12.	ARE	2	32.	THEN	1
13.	BE	1	33.	TO	6
14.	BECAUSE	1	34.	TWO	1
15.	EACH	1	35.	UNTIL	1
16.	FOR	4	36.	WAS	2
17.	INSTEAD	2	37.	WHILE	1
18.	IS	2	38.	WITH	2
19.	IT	4	39.	WITHOUT	1
20.	ITS	1	40.	WOULD	1
					99

OPERANDS

1.	APPLY	1	28.	LANGUAGE	1
2.	APPROACH	1	29.	LARGE	1
3.	ASSEMBLE	1	30.	LIMITATIONS	1
4.	ASSOCIATE	1	31.	MAP	1
5.	AVOID	1	32.	MODULE	1
6.	CERTAINLY	1	33.	MODULES	2
7.	COMPILER	4	34.	NAMES	1
8.	CONST	1	35.	NEW	1
9.	CORRECTLY	1	36.	OBVIOUS	1
10.	DECK	1	37.	PARSE	1
11.	DECLARATIONS	2	38.	PASCAL	3
12.	DESIGN	2	39.	PASSES	1
13.	DESIRABLE	1	40.	PREPROCESSOR	2
14.	DETECTED	1	41.	PRIMARILY	1
15.	DIFFICULTY	1	42.	PROGRAM	2
16.	DISADVANTAGES	1	43.	PROGRAMMERS	1
17.	DUPLICATES	1	44.	REJECTED	1
18.	ERRORS	2	45.	REORDER	1
19.	EXAMINES	1	46.	REQUEST	1
20.	FACILITY	1	47.	REQUIRES	1
21.	GENERATED	1	48.	RUN	1
22.	GROUP	1	49.	SET	1
23.	INDENTIFICATION	1	50.	SOURCE	3
24.	INCLUDE	1	51.	TIME	1
25.	INPUT	3	52.	TOOL	1
26.	INVOLVED	1	53.	TYPE	1
27.	LABEL	1	54.	VAR	1
					69

Draft Page 16

11:29:30 to 11:36:30; T = 420 sec = 7.00 min

OPERATORS

1.	#	1	18.	IT	1
2.	UC	8	19.	MORE	1
3.	.	5	20.	NOT	1
4.	,	3	21.	OF	1
5.	" "	2	22.	ONE	1
6.	ALL	1	23.	ONLY	1
7.	ALTHOUGH	1	24.	SEVERAL	1
8.	AND	1	25.	SINCE	1
9.	ARE	1	26.	SUCH	1
10.	AS	2	27.	THAT	1
11.	AT	1	28.	THE	4
12.	BE	1	29.	THEM	1
13.	BY	1	30.	THESE	1
14.	CAN	1	31.	THEY	1
15.	FOR	1	32.	THUS	1
16.	HAVE	1	33.	TO	2
17.	IS	1	34.	WELL	1
					53

OPERANDS

1.	ACCEPTS	1	18.	MAP	3
2.	ACTUALLY	1	19.	MINOR	1
3.	ADVANTAGE	1	20.	OPERATOR	1
4.	ANNOYING	1	21.	OUTPUT	1
5.	ATTEMPTS	1	22.	PASCAL	3
6.	CHANGES	1	23.	PREPROCESSED	1
7.	CHARACTER	1	24.	PRINTS	1
8.	CHARACTERS	1	25.	PROBLEMS	1
9.	COMPILER	1	26.	PROGRAM	1
10.	DETECTS	1	27.	PROGRAMS	1
11.	EASILY	1	28.	REMOVED	1
12.	EXAMPLE	1	29.	RESULTING	1
13.	EXTENSIONS	1	30.	STANDARD	3
14.	FINAL	1	31.	TIMES	1
15.	INSURMOUNTABLE	1	32.	$\leq$	1
16.	LISTING	1	33.	=	1
17.	LOCAL	1			39

Draft Page 17

11:55:40 to 12:02:25; T = 405 sec = 6.75 min

OPERATORS

1.	#	2
2.	UC	11
3.	.	5
4.	,	8
5.	'	1
6.	-	1
7.	V	1
8.	A	6
9.	ALL	1
10.	AND	3
11.	AS	1
12.	BY	1
13.	FOR	4
14.	FROM	2
15.	FURTHERMORE	1
16.	IN	1
17.	IS	3
18.	IT	1
19.	OF	3
20.	SEVERAL	1
21.	THAT	1
22.	THE	7
23.	TO	2
24.	WHICH	1
25.	WITH	1
		69

OPERANDS

1.	AID	1	27.	LARGE	1
2.	CLEARLY	1	28.	MACRO	1
3.	CODE	1	29.	MACROS	1
4.	COMPILATION	1	30.	MAINTENANCE	1
5.	COMPILE	1	31.	MAKE	1
6.	COMPILER	2	32.	MANAGEMENT	1
7.	COMPILING	1	33.	MAP	4
8.	CONCLUSIONS	1	34.	MODULAR	1
9.	CONDITIONAL	1	35.	PASCAL	2
10.	DEBUG	1	36.	PORTABLE	2
11.	DELINEATES	1	37.	PREPROCESSOR	3
12.	DESIGNED	1	38.	PROJECT	1
13.	DEPENDENT	1	39.	PROGRAMMERS	1
14.	DEVELOPMENT	2	40.	PROVIDED	1
15.	EASY	1	41.	PROVIDES	1
16.	EVALUATION	1	42.	REASONABLY	1
17.	EXPRESSION	1	43.	SINGLE	1
18.	EXTENSIONS	1	44.	SMALL	1
19.	FILE	1	45.	SOFTWARE	2
20.	FILES	2	46.	SOURCE	1
21.	FACILITIES	1	47.	SUPPORT	1
22.	HELPS	1	48.	TIME	1
23.	IMPLEMENTATION	1	49.	TOOL	1
24.	INCLUDED	1	50.	TYPICAL	1
25.	ISOLATING	1	51.	UNLIKE	1
26.	LANGUAGE	2	52.	USERS	1
					64

Draft Page 18

12:02:35 to 12:03:45; T = 70 sec = 1.17 min

OPERATORS		
1.	UC	2
2.	.	1
3.	,	1
4.	A	1
5.	ARE	1
6.	IT	1
7.	ON	1
8.	THE	1
9.	THERE	1
10.	WHEN	1
11.	WHILE	1
		12

OPERANDS		
1.	ADVANTAGES	1
2.	CLEARLY	1
3.	DISADVANTAGES	1
4.	GAINED	1
5.	INVOLVED	1
6.	LARGE	1
7.	MAKE	1
8.	MAP	1
9.	MINOR	1
10.	PROJECT	1
11.	USING	1
12.	WORTHWHILE	1
		12