Microcomputer word processing software: A functional perspective

by HAL BERGHEL
University of Arkansas
Fayetteville, Arkansas

ABSTRACT

We propose a taxonomy of features of word processing software which can be used to summarize their functional characteristics. This taxonomy is then applied to existing products in order to derive estimates of variation between products, and to extract meaningful trends.

INTRODUCTION

One of the most important objectives of a data processing manager is the selection of software which is appropriate for his/her objectives. Appropriateness, of course, is a complex objective. The manager must be sensitive to the cost-effectiveness of the product, its performance and ease of use, compatibility with products already in use, conversion time, error handling characteristics, quality of documentation, reliability, and so forth. But first and foremost, the manager must be able to determine whether the functionality of the software is adequate with respect to present and future data processing objectives. In plain terms: if the software fails to provide the range of features required by the application, its utility may be marginal.

In this paper, we propose an analysis of microcomputer word processing software which we have found useful in evaluating current products. So that no confusion results, it is useful to contrast an "analysis" with a "rating." Analyses, in our view, separate the components of the software and examine their properties and interrelationships. Ratings, on the other hand, assign values to products which purport to measure their quality. While ratings can be useful, they do have some drawbacks.

First, their value is proportional to the degree of rigor and thoroughness of the underlying methodology, which is usually not described in detail. Second, in order for any overall rating or ranking to be meaningful, the 'rater' and the user must agree with respect to the weighting schemes used (e.g., that feature 1 is as important as feature 2). Third, due to the volatility of the software industry, the useful life of the rating is very short. These first two weaknesses imply an uncertainty regarding the confidence level to assign to the report. The third weakness implies that most ratings will be obsolete before they are read.

In our opinion, the easiest way to avoid these difficulties is to provide the decision maker with the tools for analysis, rather than the results. The classification scheme presented here falls far short of perfection. However, we have found it to be a satisfactory framework for evaluation of word processing systems.

While the statistical results below focus upon microcomputer word processing systems, the analysis itself applies to word processing systems in general. We have chosen to apply the analysis to microcomputers because of their prominence in the office automation market. A brief glance through such trade publications as *Data Sources*[1] and *datapro*[2] will reveal the number of microcomputer word processing packages sold far exceeds the number of dedicated systems in use.

WORD PROCESSING ANALYSES

Our analysis works with a classification scheme for word processors which determines functionality by analyzing the command structure of the product. Since this taxonomy is the key to the analysis, some general remarks are in order.

We use the term "word processing software" to refer to a class of application programs which allows the user to create, display, edit, and store documents with a computer. No distinction is made between hardware environments, whether dedicated, stand-alone, microcomputer, or mainframe.

Word processing software, in our view, consists of five functionally distinct components: a text editor, a document manager, print, and display controllers, and a formatter. Each of these is interrelated and may be directly accessed by the user (see Figure 1). While we shall keep these components distinct in our discussion, we recognize that they may not be independent in a particular product. For example, it is not uncommon for modern products to merge the formatter and the text editor.

Each of these individual components relates to a particular class of activity involving an electronic document. By means of the document manager, the user creates, deletes, and stores such documents. Through the display and print controllers, the user exercises control over the media of presentation of the document. With the formatter, the user alters the form or appearance of the document. The text editor, however, is the kernel of the word processing software. Only the text editor supports the change of content of the document.

In our model, there is an input device (keyboard) and three peripherals (printer, display, and secondary storage). Input information falls into two categories: command information and text. Textual information is entered through the text editor, alone, whereas command information may be accepted by each subsystem. We seek to discover the functionality of a word processing system by means of a taxonomy of the commands supported. We argue that this is both a reasonable and concise description of the domain. It is a relatively objective, user-oriented, and inexpensive alternative to more complex analyses.

LEVELS OF ANALYSIS

There is no orthodoxy when it comes to the analysis of functionality of word processors. Even analyses which purport to describe word processors from the user's point of view differ significantly in terms of scope and detail. In fact, the level sometimes varies within the analysis. For example, Riddle[3] deals with such details as buffering techniques, command line

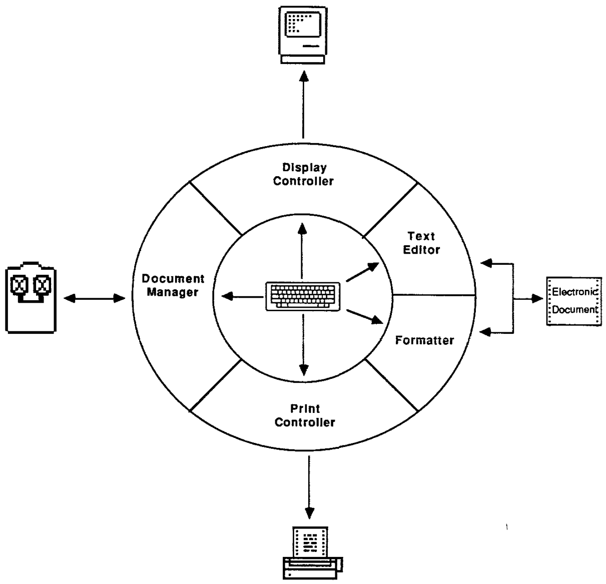

Figure 1—Five functional components of word processor

structure, and how tabs are handled but subsumes all variations of "locate" operations into one feature, and ignores the display control altogether. In this case, the analysis ranges from what Meyrowitz and van Dam[4,5] call the structural, or architectural, level to a description at the level of complex task.

Greater consistency is achieved by Roberts and Moran.[6,7,8] In this case, the analysis is usually at the level of a core editing task which is to be taken as the cross-product of a basic text editing operation (e.g., insertion and deletion) applied to basic text entities (e.g., characters, words, and sentences). This sort of analysis explicates the functionality of a product in terms of the range of core tasks supported. We have three objections to the use of core editing tasks. First, we know of no consensus regarding the appropriateness of certain core tasks to word processing applications as a whole. Failing consensus, the relevance of the core tasks to a particular work-setting must be demonstrated before the reliability of the measure may be determined. Second, all word processors, or a Turing machine, for that matter, support the same range of core tasks in one sense: it is simply a matter of how much work is involved. Thus, the issue of functionality must be further explained in terms of effort. This entails empirical study of a fast-moving and volatile market. Third, core task analyses ignore the manner of implementation of the task: generally, factors which have nothing to do with effort, but may be nonetheless, of considerable interest to the user.

For example, previewing a document with a continuous-formatting word processor is quite different than with preview-mode editors. Text editors which store control information as printable characters, as opposed to control characters not only behave differently but also generate electronic documents with distinctive properties. Similarly, insertion by split-screen might appeal to a different audience than insertion in a move-aside fashion. The point here is that the design philosophy behind a word processor is likely to affect its overall utility in ways which may not directly translate into effort. We believe that for these three reasons, at least for a first pass over the current products, the user would prefer a simpler, and less formidable, analysis than one based upon tasks.

Our study, then, attempts to extract the measure of functionality of a word processor from its command structure, including the manner of implementation of the basic operations involved, where important. We will list a feature as supported if and only if there is a specific command sequence which invokes it, or if it is a consequence of some other such sequence (e.g., automatic reformatting after deletion). In many ways, it is similar in approach to the analyses and ratings found in the popular and trade books,[9,10,11] with the exception that our taxonomy is generally of greater detail and reflects an attempt to hierarchically order the features.

PROBLEMS OF ANALYSIS

All taxonomies reflect the preferences and objectives of the taxonomist. In order to avoid inundation by detail, we have reduced the range of commands and implementation characteristics to one which we feel is both manageable and appropriate for acquisition strategies. Our selection of features is not immune to criticisms of arbitrariness. The only justification we can offer is that we have found it to be more useful than the known alternatives. Since the taxonomy serves as a filter to separate the products which match applications requirements from those which don't, the validation of the method is ultimately going to be the approbation of the user.

We specifically arranged the taxonomy to agree with what we feel are typical perceptions of features. This strategy has several implications:

1. The same underlying operation may appear as two or more separate features. This occurs when two or more commands are functionally identical, although not perceived as such. For example, some semantically simple commands (e.g., grammatically oriented operations like word and sentence deletion) are only approximated in software as special cases of other types of operations (delete-to-target). Since the user perceives the distinction between these two types of commands, the features are individuated.
2. Complex operations may be treated as simple features. To illustrate, block movement may actually be a two-stage process involving movement to and from a save buffer. In such situations, we try to orient the taxonomy toward the task rather than the method of implementation. In this case, it is our feeling that the user is more interested in adding, deleting, and permuting blocks than read/write operations on buffers.
3. A single feature may appear more than once in the taxonomy. This arises whenever a single feature affects several components of the word processor. Typographical enhancements are a paradigm case. A word processor may support boldfacing and underlining as

formatting features yet not support them on the display. Since it makes sense to speak of typographical enhancements in both contexts, they are included twice.

While complete objectivity in classification would be desirable, the complexity of current products does not allow this luxury. Our classification reflects our attitudes. Other investigators would arrive at different conclusions.

TAXONOMY OF FEATURES

The taxonomy used in this analysis appears in Appendix A. There are 168 features which break down as follows: 34 for screen control, 18 for document management, 60 for text editor, 34 for formatter and 17 for print control. In addition, there are 7 features of a more general nature (e.g., price, whether the word processor is a member of an integrated package, etc.).

Due to space considerations, we are not able to describe the features here. We only mention that we have attempted to standardize the nomenclature so that the name of the feature is descriptive of its function, without resorting to jargon. As an example, we prefer "restore text" to the often used surrogates "yankback," "undelete," and "undo," and "contextual navigation" to "find," "locate," and "search."

In addition, we provide descriptive phrases for as yet unnamed implementation characteristics. For example, "unrestricted cursor movement" refers to the ability of a word processor to move the cursor in any direction, regardless of the layout of the document. This is to be distinguished from the common restriction whereby the cursor is restricted to text boundaries as displayed. Similarly, "derivative naming convention" refers to the fact that the word processor's document manager adopts the file naming convention of the host operating system. If we have been successful, the meanings of most features are recoverable from context. Some of the concepts and terminology are covered in standard reference works.[12,13]

As we mentioned above, the taxonomy itself is the tool of the analysis. It would be used in the following way. First, the typical applications are identified. Second, the decision maker extracts from these applications prioritized sets of desirable features. Then, products are selected according to the degree to which they possess these features. For example, a boilerplating operation may find a full range of block operations indispensable, while typographical enhancements are of minimal interest. In contrast, a medical office may not need much of a formatter but a strong editor, and so forth.

While it is not possible, or even worthwhile, to present the details of the feature analysis with respect to current products, we would be remiss if we failed to provide some understanding of the current state-of-the-art. We do this in two ways. First, we compare typical microcomputer products to their dedicated counterparts. Second, we provide descriptive statistics which summarize the features of common microcomputer software. The data used are taken from 22 microcomputer word processing products marketed within the past few years. We have intentionally included the older CP/M products so that meaningful time trends can be determined. As far as practicable, we sought to include the better selling products according to *Data Sources.*[1] Products are limited to CP/M and MS-DOS, for they represent the dominant operating systems for business applications over the past decade.

MICROCOMPUTER VERSUS DEDICATED SOFTWARE

In order to place the results that follow in perspective, we conducted a comparative analysis between microcomputer-based and dedicated products. The three dedicated systems which we used (Wang WP Plus, Decmate II, and Apple's Lisa Write) were selected because of easy access and our belief that they are typical examples.

There is no doubt that, in principle, greater power can be obtained from dedicated systems than general purpose microcomputers. Dedicated systems may take advantage of all of the hardware/software integration possible, for the details of the environment are known in advance. However, we found that the three dedicated packages which we studied failed to realize this potential.

While the dedicated systems were, on average, superior in terms of both display control and document management due to the fact that the software is designed with both a specific display and operating system in mind, their advantage eroded when it came to text editing, formatting, and print control. In fact, when one considers the percentage of features supported by the two groups of software (see Table I), the microcomputer-oriented products surpassed the dedicated products overall. The mean percentages of both groups are depicted graphically in Figure 2.

It is interesting to note that the dedicated systems fell behind the microcomputer word processors with respect to what one might refer to as the more innovative features. We have in mind such things as geometrical operations (e.g., column swap), bidirectional deletion of contextual unit (e.g., sentence deletion), searches for targets consisting of non-printable characters, complex searches (e.g., searches for multiple strings), concurrent editing of multiple documents, and so forth.

Similarly, many of the more advanced formatting features (e.g., kerning, widow/orphan adjust, footnote tie-in) and print control features (e.g., chaining, merging, queuing) were frequently unsupported. In general, the dedicated software

TABLE I—Percentage of features supported by product type and functional component

MICROCOMPUTER SYSTEMS				DEDICATED SYSTEMS		
LOW	MEAN	HIGH		LOW	MEAN	HIGH
8.8	39.3	58.5	DISPLAY	47.0	52.9	55.9
22.2	44.5	72.2	DOC MAN	27.8	64.8	88.9
39.0	53.9	67.8	EDITOR	39.0	43.5	45.8
34.2	63.1	77.1	FORMATTER	45.7	54.3	65.7
43.7	66.2	81.2	PRINT CTL	31.3	43.8	68.8

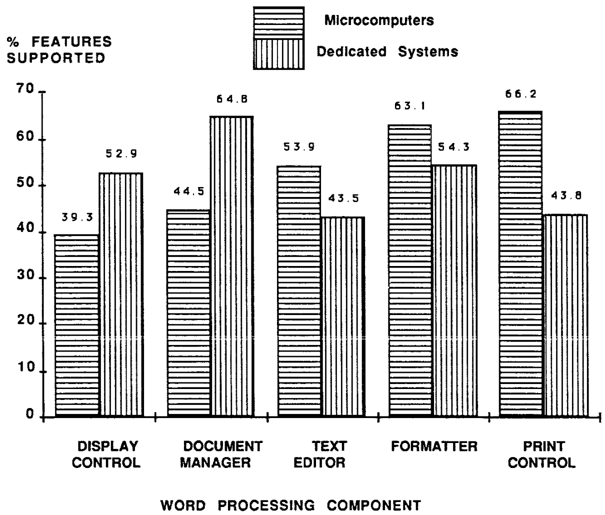

Figure 2—Mean percentage of features supported: microcomputer vs. dedicated systems

TABLE III—Mean composite scores as percentages of features

VARIABLE	MEAN	STANDARD DEV.
Display Controller	42.371%	11.989%
Document Manager	34.094%	17.856%
Text Editor	48.100%	6.889%
Formatter	52.787%	14.952%
Print Controller	60.794%	13.388%
TOTAL	47.852%	6.571%

showed clear superiority only when compared to the oldest of microcomputer-based products.

One possible explanation is that the dedicated systems are not subject to the same competitive pressures as the microcomputer software. As a result, the manufacturer may be less likely to continuously revise the product. Another contributing factor might be that manufacturers of the dedicated products rely upon the advantages of convenience, power, and speed of their system, rather than the functionality of the software. Or perhaps, the prohibitive start-up costs and limited audience discourage innovators from trying to penetrate this market. In any case, we think that this comparison indicates that the results to follow may well extend beyond microcomputer software to word processing software, generally.

DATA ANALYSIS

The 22 products studied are listed atop Appendix A. For this particular sample, 16 of the 168 variables were eliminated due to lack of variance. These variables are marked {~V} in the Appendix. Of the remaining 152 variables, 135 were dichotomous, 6 were integer and 11 were nominal (categorical). The integer and nominal variables are marked {I} and {N}. Of the 17 integer and nominal variables, all but two (date and price) were dichotomized for the analysis. According to Nie, et al.[14] dichotomies can be treated as interval-level measures. Four subordinate variables, marked {~v}, had less than 22 values because the values were dependent upon superordinate variables.

Of the 152 variables used in this analysis, 147 represented features of the five functional components of a word processor described above, and 5 were general descriptors. The dichotomous values were '0' and '1', indicating absence and presence of features in a particular product, respectively. The percentage of products which have a particular feature appear within parentheses alongside the feature entry in the Appendix. In addition, we have added 6 composite variables, one for each of the five functional components and a total, which summarize the data for each product. The descriptive statistics for the composite variables, in terms of both raw scores and percentages of possible features, appear in Tables II and III.

Pearson correlations among the 6 composite variables are given in Table IV. Since the directions of these correlations were not predicted, two-tailed tests of significance were

TABLE II—Descriptive statistics for six composite variables

VARIABLE	MEAN	STD DEV	SKEW	RANGE
Display Controller	11.864	3.357	-.751	17
Document Manager	5.455	2.857	.409	9
Text Editor	26.455	3.789	.507	14
Formatter	16.364	4.635	-.694	17
Print Controller	9.727	2.142	-.505	8
TOTAL	69.864	9.593	.020	40

TABLE IV—Pearson correlations among composite variables

	DC	DM	TE	F	PC	TOT
Display Controller		.21	.50*	.25	.04	.73*
Document Manager			-.20	.36	.11	.49*
Text Editor				.11	-.25	.50*
Formatter					.05	.73*
Print Controller						.20
Total						

* = $p < .05$

TABLE V—Pearson correlations among general variables

	OS	CP	IP	D	P
Operating System		.39*	.13	.44*	.02
Copy Protection			.33	.14	-.17
Integrated Package				-.14	.17
Date					-.57*
Price					

* = p < .05

TABLE VII—Correlations among 22 products

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
	.65	.24	.02	.14	.08	-.01	.26	-.03	.11	.30	.32	.07	.20	.13	.02	.21	.21	.21	.08	.12	.17	1
		.24	.12	.14	.21	.05	.33	.00	.17	.37	.40	.14	.14	.09	.06	.32	.35	.24	.07	.17	.19	2
			.14	.09	.32	.10	.28	.10	.18	.40	.40	.30	.27	.03	.08	.18	.40	.24	.24	.25	.17	3
				.14	.32	.34	.27	.41	.32	.07	.34	.33	.27	.35	.41	.25	.28	.35	.23	.29	.35	4
					.31	.18	.29	.17	.11	.23	.12	.23	.13	.21	.18	.09	.31	.20	.14	.28	.18	5
						.24	.46	.24	.38	.25	.46	.32	.34	.30	.35	.30	.68	.38	.35	.46	.44	6
							.12	.42	.25	.06	.16	.24	.22	.55	.56	.21	.24	.24	.21	.27	.25	7
								.30	.20	.21	.30	.27	.22	.23	.28	.33	.44	.17	.15	.22	.16	8
									.27	.00	.11	.22	.17	.43	.55	.21	.28	.26	.08	.24	.24	9
										.15	.41	.37	.19	.21	.28	.38	.42	.22	.32	.35	.53	10
											.33	.18	.20	.05	.07	.16	.31	.19	.37	.28	.26	11
												.29	.28	.27	.21	.46	.49	.24	.29	.27	.36	12
													.20	.25	.25	.19	.23	.18	.41	.33	.30	13
														.29	.22	.38	.46	.51	.30	.36	.24	14
															.78	.30	.27	.38	.15	.38	.30	15
																.21	.21	.24	.14	.38	.29	16
																	.38	.32	.21	.25	.37	17
																		.40	.32	.38	.39	18
																			.21	.36	.37	19
																				.36	.39	20
																					.47	21
																						22

used. Pearson correlations for the general variables appear in Table V. Since the directions of three correlations (operating systems and copy protection, operating systems and date, and date and price) were predicted, one-tailed tests were used. In addition, correlations between the general variables and the composite variables were determined (see Table VI) for two-tailed tests. The correlation matrix for all 22 products appears in Table VII. For this analysis, the remaining two integer variables (date and price) were dichotomized for equal weighting. Since there is a question of whether standard significance tests have any obvious meaning when cases are correlated across variables,[15] the probability values are not reported.

A cluster analysis was performed for the 22 products in order to determine similarities in functional capabilities. The clustering method used was the hierarchical, agglomerative, average-linkage between groups method provided by SPSSX, Release 2.0. We sought to avoid the extremes of single linkage clustering and complete linkage clustering. Squared Euclidean distance was employed as the proximity measure. The five general variables were not included in the clustering. Missing values for the dichotomous variables were encoded as 0.5, so that they would not be excluded by the SPSSX procedure. The dendrogram for this analysis appears as Figure 3.

TABLE VI—Pearson correlations between general and composite variables

	DISP.	DOC. MAN.	ED.	FORM.	PRT.	TOTAL
OS	.25	.17	.002	.006	.02	.15
CP	.06	-.03	-.15	.12	.06	.02
IP	-.08	-.10	-.23	.08	-.20	-.16
DATE	.17	.17	-.32	-.27	.44*	-.05
PRICE	.09	-.12	.17	.53*	-.27	.26

* = P < .05

DISCUSSION

Several observations can be made from the descriptive statistics in Tables II and III. First, The document manager is clearly the weakest part of these packages. This can be confirmed by reference to the Appendix. Only 18.2% of the products tested supported a document naming convention which was independent of the host operating system. In a CP/M and MS-DOS environment, this means that all file names must conform to the 8 character name/3 character extension format. Further, less than one third of the products allowed the user to identify the document by author, owner, and dates of creation and revision. Perhaps the most striking weakness, however, is the fact the less than one fourth of the products had document managers which were consistent with the file management facilities of the host operating systems. In most cases, this was a result of MS-DOS products failing to support the tree-structured domain supported in versions 2.X and above. This means that it is not possible to define multiple directories according to type of document within the word processor.

At the other extreme is the print controller. Its overall strength might suggest that software houses are investing a

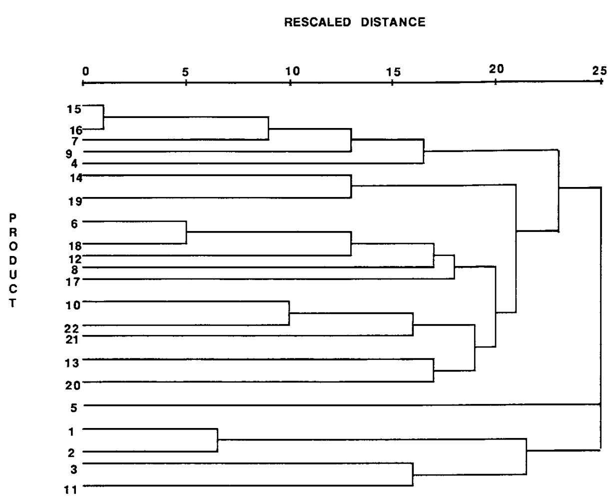

Figure 3—Cluster dendrogram

great deal of time and effort in this direction. In fact, when one looks at the correlations in Table IV, one sees that print control is significantly and strongly correlated with date. Apparently, print control is perceived to be an important component, worthy of continued attention.

Another observation is that the standard deviation in percentage of features of the "total" composite variable is less than the standard deviations for each individual composite variable (see Table III). It would appear that weaknesses in one component are generally compensated for by other components. Perhaps, as an overall design philosophy, software houses are trying to appeal to specific audiences by focusing on their particular needs. Another possibility is that they continue to refine those components with which they have the most experience. A third possibility is that software houses include features in a somewhat random fashion. This last possibility is not inconsistent with the Pearson correlations between the composite variables (Table IV). Generally, only the total composite variable is strongly and significantly correlated with the individual composite variables, which is understandable since the latter make up the former. The lack of correlation between the individual composite variables indicates that the five functional components of word processors are largely independent of one another: strength in one area says nothing of another. The exception to this rule is that the quality of screen display tends to correlate with the quality of the text editor.

Table V was created to determine the accuracy of our intuitions. It should be no surprise to anyone that copy protection is primarily associated with MS-DOS products, that the newer products tend to be designed for MS-DOS, and that the price of products is decreasing over time. We suspect that if this study had been conducted a few years ago, we would have found a positive, statistically significant correlation between date and copy protection, as well. However, this practice has recently fallen into disrespect.

The correlation of the five general variables with the composite variables (see Table VI) is strongly suggestive of some overall patterns. First, as we mentioned above, print controllers seem to be getting stronger over time. Further, price seems to be a good indication of the quality of the formatter, though little else. If one looks to the Pearson correlations between price and the individual features (not shown), one finds that it is positively correlated with only automatic hyphenation (.48, $p < .05$), the presence of screen labelling of function keys (.5, $p < .05$), whether page numbering can be included in headers and footers (.49, $p < .05$), and the capability of double striking (.49, $p < .05$). Specifically, price is not shown to correlate well with such advanced features as multiple windowing and integrated graphics. What this tells us is that price is not a measure of the overall quality of the product. Similarly, the lack of positive correlation between copy protection and the composite variables suggests that it is unlikely that there is any relation between the quality of a product and copy protection. In terms of our sample, copy protection seems to have had the unintended effect of discouraging piracy of the poorer products.

In addition, our experience indicates that the negative correlations between most of the composite variables and the general feature, membership in an integrated package, seem reasonable. Again, we predict that if a large enough sample is taken, these negative correlations can be shown to be statistically significant. We suspect that the integration comes at the expense of the quality of the subcomponents.

Table VII and Figure 3 are useful in identifying relationships between products. For example, there are three pairs of word processors which bear the same name: Perfect Writer, Benchmark and Easywriter. In one case, Perfect Writer, the CP/M and MS-DOS versions are seen to be functionally similar. Slightly less similar, though related by philosophy, are the Benchmarks. However, the Easywriters are essentially two different products. In addition, one gets the feeling the Freestyle and Select, and Newword and Wordstar have some common ancestry. This information may be of use to decision makers who would like to examine products with similar orientations.

CONCLUSIONS

In this paper, we have presented a taxonomy of features of word processing features which we have found to be useful in determining the functionality of word processing software and the relevance of software to particular applications. In addition, we have provided summary statistics for 22 existing products, when compared in terms of the taxonomy. It is our intention to provide the reader with a means of identifying distinctive and distinguishing features, as well as provide some method of assessing the microcomputer word processing software market as a whole. We hope that this information is useful in aiding decision makers who are interested in acquiring new software.

While space constraints dictate that we omit a general discussion of the correlations between features, we would like to conclude with some preliminary observations. First, the correlations indicate that innovation in word processor design is likely to be a random event. To illustrate, there is no correlation among our 22 products between multiple windowing, unrestricted text entry, complex searches, concurrent editing, and integrated graphics—features we believe are indicative of sophistication. Further, one gets the feeling that most current products suffer from a serious lack of forethought. For example, multiple windowing negatively correlates with the quality of status line information concerning the position of the cursor in the various documents in a statistically significant way. This means that for most products that support multiple windowing, the user is left in the dark concerning the location of the window. Another example is the lack of correlation between contextual deletion (delete sentence and paragraph) and delete-to-target. Since the latter is the lower-level technique used to implement the former, its absence at the command level is strange, indeed. A third case involves the frequent failure of unrestricted text entry to accompany unrestricted cursor movement. The consequence of this is that while the user may directly navigate the cursor to any position on the screen, he may not be able to do any editing once it is there. Hopefully, with further study we may better understand these design philosophies.

ACKNOWLEDGEMENT

We would like to thank D. Lavelle for assistance with the statistical portions of this paper.

REFERENCES

1. *Data Sources,* New York: Ziff-Davis Publishing Co.
2. *datapro directory of MICROCOMPUTER SOFTWARE,* Delran, NJ: McGraw-Hill.
3. Riddle, E. "Comparative Study of Various Text Editors and Formatting Systems." Air Force DataServices Center Report #AD-A-29 050, 1976.
4. Meyrowitz, N. and A. van Dam. "Interactive Editing Systems." *Computing Surveys,* 14 (1982) 3, pp. 321–415.
5. van Dam, A. and N. Meyrowitz. "Text Editing Systems." A. Ralston and E. Reilly, Jr. *Encyclopedia of Computer Science and Engineering* (2nd ed.). New York: van Nostrand Reinhold, 1983.
6. Roberts, T. "Evaluation of Computer Text Editors." PhD Dissertation, Stanford University, November, 1979
7. Roberts, T., and T. Moran. "Evaluation of Text Editors." *Proceedings of the Conference on Human Factors in Computer Systems,* March 1982, pp. 136–141.
8. Roberts, T., and T. Moran. "The Evaluation of Text Editors: Methodology and Empirical Results." *Communications of the ACM,* 26 (1983) 4, pp. 265–283.
9. Smith, B. and D. Austin. *Word Processing: A Guide for Small Business,* Brattleboro: Lewis Publishing, 1983.
10. Naiman, A. *Word Processing Buyers Guide.* New York: McGraw-Hill, 1983.
11. Donahue, B. *How to Buy an Office Computer or Word Processor.* Englewood Cliffs: Prentice-Hall, 1983.
12. Flores, I. *Word Processing Handbook.* New York: van Nostrand, 1983.
13. Sampath, G. *An Introduction to Text Processing.* Jeffersontown: River Valley, 1985.
14. Nie, N., C. Hull, J. Steinbrenner and D. Bent. *Statistical Package for the Social Sciences* (2nd ed.). New York: McGraw-Hill, 1975.
15. Aldenderfer, M. and R. Blashfield. *Cluster Analysis.* Beverly Hills: Sage Publications, 1984.

APPENDIX A: TAXONOMY OF WORD PROCESSING FEATURES

Products Tested:

Product (1) Benchmark (CP/M), v. 3.0m
Product (2) Benchmark (DOS), v. exec III
Product (3) DisplayWrite 2 (DOS), v. 1.10
Product (4) Easywriter I (DOS), v. 1.40
Product (5) Easywriter II (DOS), v. 2.0
Product (6) Freestyle (DOS), v. 1.0
Product (7) Final Word (DOS), v. 1.17
Product (8) Leading Edge (DOS), v. 1.20
Product (9) Memoplan (CP/M), v. 1.2
Product (10) Newword (CP/M), v. 1.29
Product (11) Office Writer (DOS), v. 3.0
Product (12) Palantir (CP/M), v. 1.2
Product (13) pfs:Write (DOS), v. b
Product (14) Peachtext (CP/M), v. 2.01
Product (15) Perfect Writer (CP/M), v. 1.20
Product (16) Perfect Writer (DOS), v. 2.0
Product (17) Spellbinder (DOS), v. 5.30
Product (18) Select (CP/M), v. 3.00c
Product (19) Superwriter (CP/M), v. 1.02
Product (20) Visiword (DOS), v. 1.20
Product (21) Volkswriter (DOS), v. 2.10
Product (22) Wordstar (CP/M), v. 3.32

Taxonomy:

0. GENERAL INFORMATION
    —Version
    —Operating System
    —Copy Protected
    —Member of Integrated Package
    —Date of Release {I}
    —Price {I}

I. DISPLAY
  A. Screen Imaging
    1. Layout
      —centering (100%) {−V}
      —justification (68.2)
      —line spacing (40.9)
      —pagination (77.3)
      —hyphenation (45.5)
    2. Typography
      —boldface (22.7)
      —sub/superscripts (0.0) {−V}
      —strikeouts (4.5)
      —underlining (45.5)
      —overprints (0.0) {−V}
      —alternate fonts (0.0) {−V}
      —alternate pitch (0.0) {−V}
      —proportional spacing (9.1)
    3. Control Suppression (50.0)
  B. Highlighting
    —block (63.6)
    —character (59.1)
  C. Column Ruler Display (77.3)
    —on/off (18.2)
    —multiple rulers (40.9)
  D. Status Line Display
    1. Document ID
      —drive id (59.1)
      —document name (77.3)
    2. Cursor Location
      —page number (59.1)
      —line number (77.3)
      —column number (63.6)
    3. Editor Mode Toggles (90.9)
    4. Systems Information
      —document size (4.5)
      —remaining space on disk (13.6)
      —remaining space in RAM (18.2)
      —time/date (9.1) {N}
      —screen labelled function keys (22.7)
  E. Multiple Windowing (31.8)
    —number of windows (mean = 2) {−V}
    —number of documents (mean = 4.5) {I} {−v}
    —variable size (18.2) {−v}

II. DOCUMENT MANAGEMENT
- A. Document Naming Convention
  - —derivative/independent (18.2% = independent)
  - —maximum character length (mean = 12) {I}
  - —'wild card' reference (40.9)
- B. Backup
  - —manual/auto (63.6% = manual) {N}
  - —override (95.5) {−v}
- C. Directory
  - —document size (50.0) {N}
  - —description (18.2)
  - —author (22.7)
  - —creator (22.7)
  - —date created (31.8)
  - —last revision (31.8)
  - —total worktime (0.0) {−V}
  - —archive reference (0.0) {−V}
- D. Constraints and Security
  - —document size (77.3)
  - —access consistent with OS (22.7)
  - —edit-protect (9.1)
  - —exit protection (31.8) {N}
  - —disk overflow protection (77.3) {N}

III. TEXT EDITOR
- A. ADD TEXT (Insert)
  1. Move Aside (95.5)
     - —word wrap/cascading wraparound (27.3% + cascading)
  2. Split Screen (45.5)
  3. Unrestricted Text Entry (27.3)
- B. DELETE TEXT (Delete)
  1. Units
     - a. Contextual
       - —character (95.5)
       - —word (63.6)
       - —sentence (27.3)
       - —paragraph (27.3)
     - b. Geometrical
       - —columns (27.3)
       - —lines (72.7)
     - c. Boundary
       - —delete to end (45.5)
       - —delete to target (22.7)
       - —delegate marked block (95.5)
  2. Miscellaneous
     - —bidirectional deletion (27.3)
     - —deletion w/prompt (59.1)
     - —restore (68.2)
     - —variable save space (9.1)
     - —auto-reformat (63.6)
- C. NAVIGATION (Move/Find)
  1. Geometrical
     - a. Relative location
       - —Directional
         - —unrestricted cursor movement (50.0)
         - —screen advance (95.5)
         - —scrolling
           - —vertical/horizontal (54.5/18.2) {N}
           - —hard/soft (0.0 = soft) {−V}
           - —variable speed (18.2) {−v}
       - —Grammatical
         - —word (68.2)
         - —sentence (27.3)
         - —paragraph (22.7)
     - b. Absolute location
       - —top of document (77.3)
       - —bottom of document (77.3)
       - —page by number (36.4)
       - —marker (31.8)
  2. Contextual (Locate/Find/Search)
     - a. restrictions on target
       - —length (mean = 58) {I}
       - —control characters allowed (81.8)
       - —ambiguous character strings (27.3)
     - b. restrictions on search
       - —auto-repeat (95.5)
       - —complex search (4.5)
     - c. search parameters
       - —complete words (45.5)
       - —reverse search (54.5)
       - —global search (22.7)
       - —ignore case (63.6)
- C. SUBSTITUTE TEXT (SEARCH & REPLACE)
  - a. restrictions on target
    - —length (mean = 58) {I}
    - —control characters allowed (77.3)
  - b. restrictions on search
    - —auto-repeat (100) {−V}
    - —complex search (4.5)
  - c. search parameters
    - —complete words (45.5)
    - —reverse search (36.4)
    - —global search (31.8)
    - —ignore case (63.6)
- E. PERMUTE TEXT (Block Move/Copy)
  1. Contextual Permutation (95.5)
  2. Geometrical Permutation (31.8)
     - —column swap (9.1)
  3. Options
     - —block move (100) {−V}
     - —block copy (95.5)
     - —block delete (100) {−V}
     - —block file (77.3)
- F. MISCELLANEOUS
  1. Menu Type (90.9 = pass through, remaining = pop up)
     - —variable help level (pass through only) (36.4)
     - —menu delay (pass through only) (18.2)
     - —menu bypass (pass through only) (36.4)
  2. Concurrent Editing (31.8)
  3. Integrated Graphics (9.1)

IV. FORMATTING
- A. CONTINUOUS/PREVIEW MODE (72.7 = continuous)

B. LAYOUT
1. Line Centering (100) {−**V**}
2. Variable Line Spacing (32) {**N**}
3. Proportional Spacing (72.7)
4. Kerning (9.1)
5. Justification (95.5)
—fixed/variable spacing (31.8 = variable) {**N**}
—interword (100) {−**V**}
—intraword (13.6)
6. Hyphenation (63.6)
—concurrent (50.0)
—automatic (27.3)
7. Decimal Alignment (54.5)
8. Pagination
—pagination/repagination (95.5)
—page numbering (100) {−**V**}
—with initialization ≠ 1 (81.8)
—widow/orphan adjust (63.6) {**N**}
—header/trailer insert (95.5)
—page-number merge (86.4)
—footnote tie-in (31.8)

C. TYPOGRAPHY
1. Character Enhancements
—boldface (95.5)
—complementary overprinting
—double-strike (45.5)
—underlining (100) {−**V**}
—destructive overprinting
—strikeout (59.1)
—typeover (13.6)
2. Miscellaneous
—sub/super scripts (90.0)
—multiple fonts (45.5)
—multiple character sets (36.4)
—print pause (90.9)
—print phantom character (31.8)
—multiple pitches (77.3)
—ribbon color change (22.7)
—user-definable commands (22.7) {**N**}
—typethrough (4.5)

V. PRINT CONTROL
—multiple copies (90.9)
—selective output (95.5)
—multiple pages (22.7)
—first/last page (86.5) {**N**}
—draft quality only (45.5)
—dual column printing (13.6)
—printer select (72.7)
—paper change pause (100) {−**V**}
—form feeds (90.9)
—disk file output (59.1)
—chaining (45.5)
—merging (45.5)
—queuing (36.4)
—print from edit (40.9)
—print while editing (50.0)
—print-time commands
—print stop (90.9)
—print pause/resume (86.4)